# Extending Answer Sets for Logic Programming Agents

M. De Vos (mdv@cs.bath.ac.uk)
*Dept of Computer Science, University of Bath, Bath, BA2 7AY, United Kingdom*

D. Vermeir (dvermeir@vub.ac.be)
*Dept of Computer Science, Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussel, Belgium*

**Abstract.** We present systems of logic programming agents (LPAS) to model the interactions between decision-makers while evolving to a conclusion. Such a system consists of a number of agents connected by means of unidirectional communication channels. Agents communicate with each other by passing answer sets obtained by updating the information received from connected agents with their own private information. We introduce a credulous answer set semantics for logic programming agents. As an application, we show how extensive games with perfect information can be conveniently represented as logic programming agent systems, where each agent embodies the reasoning of a game player, such that the equilibria of the game correspond with the semantics agreed upon by the agents in the LPAS.

**Keywords:** answer set programming, multi-agent sytems, knowlegde representation, game theory.

**AMS Classification:** 68T27, 68T28

## 1. Introduction

In this paper we present a formalism for systems of logic programming agents. Such systems are useful for modeling decision-problems, not just the solutions of the problem at hand but also the evolution of the beliefs of and the interactions between the agents.

A system of logic programming agents consists of a set of agents connected by means of unidirectional communication channels. Each agent contains an ordered choice logic program [12] representing her personal information and reasoning skills. Agents use information received from their incoming channels as input for their reasoning, where received information may be overridden by other concerns represented in their programs. The resulting model is communicated to the agents listening on the outgoing channels. The semantics of the whole system corresponds to a stable situation where no agent needs to change its output. To model a single agent's reasoning, we use ordered choice logic programs [12], an extension of logic programming that provides facilities for the direct representation of preference between rules and dynamic choice between alternatives. Unlike for other preference-based extensions (see Section 6), alternatives in ordered choice programs, and the related notion of defeat, are not based on negation (which would yield a static approach), but depend on the interpretation at hand which may dynamically activate inconsistencies between atoms. Nevertheless, negation (both classical and negation as failure) can be easily simulated (see Section 6.1).

Game theory [22] makes contributions to many different fields. In particular, there is a natural connection with multi-agent systems. In this paper we illustrate the use of logic programming agent systems as convenient executable representations of games, where each player corresponds to exactly one agent. We concentrate on so-called extensive games with perfect information: a sequential communication structure of players taking decisions, based on full knowledge of the past. We demonstrate that such games have a constructive and intuitive translation to logic programming agent systems where the agents/players are connected in a cyclic communication structure. The game's equilibria (Nash or subgame perfect, depending on the transformation used to construct the corresponding system) can then be retrieved as the system's answer set semantics. Moreover, its fixpoint computation closely mirrors the actual reasoning of the players in reaching a conclusion corresponding to an equilibrium.

All proofs can be found in the Appendix.

## 2. Choice Logic Programming

Choice logic programs [9, 10] represent decisions by interpreting the head of a rule as an exclusive choice between alternatives.

Formally, a *Choice Logic Program* [10], CLP for short, is a countable set of rules of the form $A \leftarrow B$ where $A$ and $B$ are finite sets of ground atoms. Intuitively, atoms in $A$ are assumed to be xor'ed together while $B$ is read as a conjunction (note that $A$ may be empty, i.e. constraints are allowed). The set $A$ is called the head of the rule $r$, denoted $H_r$, while $B$ is its body, denoted $B_r$. In examples, we often use "$\oplus$" to denote exclusive or, while "," is used to denote conjunction.

The *Herbrand base* of a CLP $P$, denoted $\mathcal{B}_P$, is the set of all atoms that appear in $P$. An *interpretation* is a consistent[1 2] subset of $\mathcal{B}_P \cup \neg\mathcal{B}_P$. For an interpretation $I$, we use $I^+$ to denote its positive part, i.e. $I^+ = I \cap \mathcal{B}_P$. Similarly, we use $I^-$ to denote the negative part of $I$, i.e. $I^- = \neg(I \cap \neg\mathcal{B}_P)$. An atom $a$ is *true* (resp. *false*) w.r.t. to an interpretation $I$ for a CLP $P$ if $a \in I^+$ (resp. $a \in I^-$). An interpretation is *total* iff $I^+ \cup I^- = \mathcal{B}_P$. The positive complement of an interpretation $I$, denoted $\overline{I}$, equals $\mathcal{B}_P \setminus I^+$.

A rule $r$ in a CLP is said to be *applicable* w.r.t. an interpretation $I$ if $B_r \subseteq I$. Since we are modeling choice, we have that $r$ is *applied*[3] when $r$ is applicable

[1] For a set of literals $X$, we use $\neg X$ to denote $\{\neg a \mid a \in X\}$, where $\neg\neg a = a$ for any atom $a$. $X$ is consistent iff $X \cap \neg X = \emptyset$.

[2] In this paper, we use $\neg$ to denote negation-by-failure.

[3] For a a set $X$, we use $|X|$ do denote its cardinality.

and $|H_r \cap I| = 1$. A rule is *satisfied* if it is applied or not applicable. A *model* is defined in the usual way as a total interpretation that satisfies every rule. A model $M$ is said to be *minimal or stable* if there does not exist a model $N$ such that $N^+ \subset M^+$.

EXAMPLE 1 (Eternal Enemies). *One day two eternal enemies, the lions and the hyaena's, meet on the African plains. Today's cause of argument is a juicy piece of meat. Both the lions and the hyaena's are keen on devouring it. To obtain their share there are two possibilities: they can either divide the piece or they can fight for it with the risk of getting injured. Both parties know that they get the most meat without any risk if they are both willing to share. Despite this, no pride is willing to take the risk of losing out on this free lunch. The following simple choice logic program models this eternal feud.*

$$
\begin{array}{rcll}
share_{lions} & \oplus & fight_{lions} & \leftarrow \\
share_{hyaenas} & \oplus & fight_{hyaenas} & \leftarrow \\
& & fight_{lions} & \leftarrow \ share_{hyaenas} \\
& & fight_{lions} & \leftarrow \ fight_{hyaenas} \\
& & fight_{hyaenas} & \leftarrow \ share_{lions} \\
& & fight_{hyaenas} & \leftarrow \ fight_{lions}
\end{array}
$$

*The program from Example 1 has one stable model $\{fight_{lions}, fight_{hyaenas}\}$, which explains why the two species remain enemies: neither wants to give sharing a try as they fear that the other will take advantage by attacking.*

## 3. Ordered Choice Logic Programming

An ordered choice logic program (OCLP) [11] is a collection of choice logic programs, called components, each representing a portion of information. The relevance or preciseness of each component with respect to the other components is expressed by a strict pointed partial order[4].

DEFINITION 1. *An **Ordered Choice Logic Program**, or OCLP, is a pair $P = \langle \mathcal{C}, \prec \rangle$ where $\mathcal{C}$ is a finite set of choice logic programs, called **components**, and "$\prec$" is a strict pointed partial order on $\mathcal{C}$. We use $P^\star$ to denote the CLP obtained from P by joining all components, i.e. $P^\star = \cup_{c \in \mathcal{C}} c$. For a rule $r \in P^\star$, $c(r)$ denotes the component from which the rule was taken (i.e. we assume that rules are labeled by the component)[5]. The Herbrand base of an*

[4] A relation $<$ on a set $A$ is a strict partial order iff $<$ is anti-reflexive, anti-symmetric and transitive. $<$ is pointed if there is an element $a \in A$ such that $a < b$ for all $b \in A \setminus \{a\}$.

[5] In fact, the same rule could appear in two components and thus $P^\star$ should be a set of labeled rules. We prefer to work with the present simpler notation and note that all results remain valid in the general case.

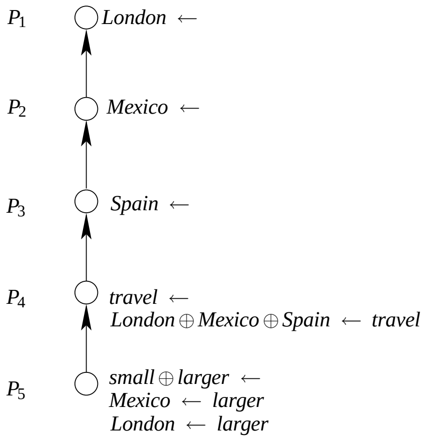


*Figure 1.* The travel OCLP of Example 2

*OCLP $P$ is defined by $\mathcal{B}_P = \mathcal{B}_{P^\star}$. An* ***interpretation*** *of $P$ is an interpretation of $P^\star$. A rule $r$ in an OCLP $P$ is applicable, resp. applied, w.r.t. an interpretation $I$, if it is applicable, resp. applied in $P^\star$, w.r.t. $I$.*

For two components $C_1, C_2 \in \mathcal{C}$, $C_1 \prec C_2$ implies that $C_2$ contains more general, or less preferred, information than $C_1$. Throughout the examples, we will often represent an OCLP $P$ by means of a directed acyclic graph in which the nodes represent the components and the arcs the $\prec$-relation, where arcs point from more to less preferred components and represent the transitive reduction of the $\prec$-relation.

EXAMPLE 2. *This year, the choice for the holidays has been reduced to a city trip to London or a fortnight stay in either Spain or Mexico. A city trip to London is rather short and Mexico is expensive. With a larger budget however, we could have both a holiday in Mexico and a trip to London. Given these considerations, there are two possible outcomes: either we have a small budget and we should opt for Spain, or with a larger budget, we can combine Mexico and London.*

*This decision problem can be conveniently represented as an OCLP, as displayed by Figure 1. The rules in the components $P_1 \ldots P_3$, express the preferences in case of a small budget. The rules in $P_4$ explain that we want to travel and, because of this, we need to make a decision concerning our destination. In component $P_5$, the first rule states that there is also the possibility of a larger budget. In this case, the two other rules in this component tell us that we can have both London and Mexico. The following sets are interpretations for this OCLP:*

– $I = \{Mexico, small, \neg Spain\}$,

– $J = \{travel, Mexico, small, \neg London, \neg Spain, \neg larger\}$,

– $K = \{travel, Spain, small, \neg larger, \neg London, \neg Mexico\}$, *and*

– $L = \{travel, London, Spain, Mexico, larger, \neg small\}$

*The interpretation $I$ makes the rule $small \oplus larger \leftarrow$ applied while the rule $London \leftarrow$ is applicable but not applied. While $J$, $K$ and $L$ are total, $I$ is not.*

When it is clear from the context that only total interpretations are considered, we will omit the negative part.

A decision involves a choice between several alternatives. In a CLP, decisions are generated by so-called *choice rules*, i.e. rules with multiple head atoms. Thus, for an interpretation $I$ of a CLP, $a$ and $b$ are alternatives if they appear together in the head of an applicable rule. For an ordered program, we will use a similar notion which takes the preference order into account. Intuitively, for a component $C$, $a$ and $b$ are alternatives w.r.t. an interpretation $I$ if there is an applicable choice rule containing $a$ and $b$ in the head, in a component that is at least as preferred as $C$.

DEFINITION 2. *Let $P = \langle \mathcal{C}, \prec \rangle$ be an OCLP, let $I$ be an interpretation of $P$ and let $C \in \mathcal{C}$. The set of* ***alternatives*** *in $C$ for an atom $a \in \mathcal{B}_P$ w.r.t. $I$, denoted $\Omega^I_C(a)$, is defined as*[6]*:*

$$\Omega^I_C(a) = \{b \mid \exists r \in P^\star \cdot c(r) \preccurlyeq C \ \wedge\ B_r \subseteq I \ \wedge\ a, b \in H_r \text{ with } a \neq b\} \ .$$

EXAMPLE 3. *Reconsider the interpretations $I$ and $J$ from Example 2. The alternatives for Mexico in $P_2$ w.r.t. $J$ are $\Omega^J_{P_2}(Mexico) = \{Spain, London\}$. With respect to $I$ we obtain $\Omega^I_{P_2}(Mexico) = \emptyset$, since the choice rule in $P_3$ is not applicable. When we take $P_5$ instead of $P_2$, we obtain w.r.t. $J$: $\Omega^J_{P_5}(Mexico) = \emptyset$.*

Atoms that are each others' alternative w.r.t. a certain interpretation $I$ will continue to be so in any extension $J \supseteq I$. In this sense, $\Omega_P$ is a monotonic operator.

Although rules do not contain negations, they can still conflict. E.g. one rule could force a choice between $a$ and $b$ while other rules could force $a$ and $b$ separately. More generally, a conflict exists for a rule $r$, which is applicable w.r.t. an interpretation $I$, if for all $a \in H_r$, there exists another rule $r_a$ such that $H_{r_a} \subseteq \Omega^I_{c(r)}(a)$. As in [17], we use the preference relation among the components to ignore rules that are *defeated* by rules that are not less preferred, forcing different alternatives.

[6] $\preccurlyeq$ is the reflexive closure of $\prec$.

DEFINITION 3. *Let $I$ be an interpretation of an OCLP $P$. A rule $r \in P^\star$ is* ***defeated*** *w.r.t. $I$ iff*

$$\forall a \in H_r \cdot \exists r' \in P^\star \cdot c(r) \not\prec c(r') \ \wedge \ r' \textit{ is applied w.r.t. } I \ \wedge \ H_{r'} \subseteq \Omega^I_{c(r)}(a) \ .$$

*The rule $r'$ is called a* ***defeater*** *w.r.t. $I$. $I$ is a* ***model*** *of $P$ iff every rule in $P^\star$ is either not applicable, applied or defeated w.r.t. $I$. A model $M$ is* ***minimal*** *iff no model $N$ of $P$ exists such that $N^+ \subset M^+$.*

The above definition is credulous in the sense that a decision is made even if the alternatives are equally preferred or unrelated. In a skeptical approach [13], one would demand that a defeater is strictly preferred over the defeated rule, i.e. $c(r') \prec c(r)$.

EXAMPLE 4. *Reconsider the interpretations $J$ and $L$ defined in Example 2. The rule $London \leftarrow$ is defeated w.r.t. $J$ by the rule $Mexico \leftarrow$. The combination of the rules $Mexico \leftarrow larger$ and $London \leftarrow larger$ defeats the rule $London \oplus Mexico \oplus Spain \leftarrow$ w.r.t. $L$. Only $K$ and $L$ are models. $L$ is not minimal due to the smaller model $Z = \{travel, larger, Mexico, London, travel, \neg Spain, \neg small\}$. The minimal models $K$ and $Z$ correspond to the intuitive outcomes of the problem.*

For ordered programs, the minimal semantics sometimes yields unintuitive results, as demonstrated in the following example.

EXAMPLE 5. *Consider the program $P = \langle \{c_1, c_2, c_3\}, \prec \rangle$ where $c_1 = \{a \leftarrow \}$, $c_2 = \{b \leftarrow\}$, $c_3 = \{a \oplus b \leftarrow c\}$ and $c_3 \prec c_2 \prec c_1$. The minimal models are $\{a, b\}$, where no choice between $a$ and $b$ is forced, and $\{c, b\}$. The latter is not intuitive due to the gratuitous assumption of $c$.*

Unwarranted assumptions as in Example 5 can be avoided by adopting an answer set semantics, employing a reduction technique as in [19] to filter out minimal models with undesirable assumptions.

DEFINITION 4. *Let $I$ be an interpretation of an OCLP $P$. The* ***reduct*** *of $P$ w.r.t. $I$, denoted $P^I$, is the choice logic program obtained from $P^\star$ by deleting every defeated rule w.r.t $I$.*
*A total interpretation $M$ is a* ***answer set*** *for $P$ iff $M$ is a stable model of the CLP $P^M$.*

EXAMPLE 6. *The program $P$ from Example 5 does not admit $N = \{c, b\}$ as an answer set, since $P^N = \{b \leftarrow, \ a \oplus b \leftarrow c\}$ which has only $\{b\} \neq N$ as a minimal model. $M = \{a, b\}$ is an answer set because $P^M = P^\star$ has a minimal model $M$. The minimal models $K$ and $Z$ of Example 4 are both answer sets.*

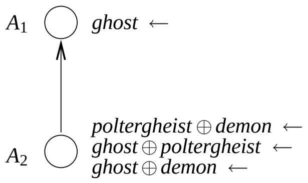


*Figure 2.* The ghost OCLP of Example 8

Obviously, the answer set semantics is a generalization of the stable model semantics for choice logic programs, where a CLP can be seen as an OCLP with a single component.

THEOREM 1. *Let $P$ be a choice logic program. $M$ is a stable model for $P$ if $M$ is an answer set of the corresponding OCLP $P_o = \langle\{P\}, \emptyset\rangle$.*

The reverse of the above theorem does not hold, as demonstrated by the following simple example.

EXAMPLE 7. *Consider the following simple set of rules:*

$$\begin{array}{rl} a \oplus b & \leftarrow \\ a & \leftarrow \\ b & \leftarrow \end{array}$$

*Considered as the rules of a CLP, we do no obtain any stable model. The last two rules make that the first can never be satisfied. On the other hand, if one considers these rules to belong to an OCLP with a single component, the set $\{a,b\}$ comes up as an answer set. This result is due to the two last rules defeating the first, overruling the exclusive decision.*

As for extended disjunctive logic programs [19], an answer set of an OCLP is not necessarily minimal.

EXAMPLE 8. *Consider the OCLP depicted in Figure 2 modeling the considerations of an exorcist. The interpretations:*

$$\begin{array}{l} I = \{ghost, \neg poltergeist, \neg demon\} \ , \\ J = \{ghost, poltergeist, \neg demon\} \end{array}$$

*are both answer sets of the Ghost OCLP with the reducts as shown below.*

$$\begin{array}{llcll} & P^I & & & P^J \\ & & & & \\ ghost & \leftarrow & & poltergeist \oplus demon & \leftarrow \\ ghost \oplus poltergeist & \leftarrow & & ghost \oplus demon & \leftarrow \\ ghost \oplus demon & \leftarrow & & & \end{array}$$

Although answer sets are in general not minimal, they are still models.

THEOREM 2. *Let $M$ be an answer set of an OCLP $P$. Then, $M$ is a model of $P$.*

In [11], an algorithm is provided for the computation of the answer sets of OCLP, using the implicit negation present in the program.

## 4. Logic Programming Agents

In this section we consider systems of communicating agents where each agent is represented by an OCLP that contains knowledge about itself and other agents.

Agents communicate via unidirectional communication channels through which the conclusions derived by the agent at the source of the channel are passed on to the agents at the other end.

DEFINITION 5. *A **logic programming agent system**, or LPAS, is a pair $F = \langle \mathbf{A}, C\rangle$ where $\mathbf{A}$ is a set of agents $a$ and $C \subset \mathbf{A} \times \mathbf{A}$ is a relation representing the communication channels between agents. Moreover, each agent $a \in \mathbf{A}$ is associated with an ordered choice logic program $F_a = \langle \mathcal{C}_a, \prec_a\rangle$.*

We will use a more convenient graph-like notation in our examples.

EXAMPLE 9. *Two witnesses discover a body lying in the park. The first witness tells the local police that she saw hair near the victim and that she did not see any blood. The second witness testifies that she saw blood and that the victim had strange bite marks. The sheriff states that this situation is a clear case of murder and passes it to the FBI. Because of the strange appearance of bite-marks and hair, the FBI passes the case to the special X-cell. In addition, the FBI states that, if the X-cell reports that a werewolf is involved, the case should be classified. Given the evidence, the X-file team has no choice but to decide that the killing was indeed done by a werewolf. This situation is represented by the LPAS depicted in Figure 3.*

The Herbrand base of a LPAS is the union of all the Herbrand bases of the ordered choice logic programs used by the agents. An interpretation assigns a set of literals to each agent in the system. These literals may be concluded by the agent itself, based on input received through an input channel, or they may simply be accepted from other agents via an input channel.

DEFINITION 6. *Let $F = \langle \mathbf{A}, C\rangle$ be an LPAS. The **Herbrand base** of $F$, denoted $\mathcal{B}_F$, equals $\mathcal{B}_F = \bigcup_{A \in \mathbf{A}} \mathcal{B}_A$. An **interpretation** of $F$ is a function $I : \mathbf{A} \rightarrow 2^{(\mathcal{B}_F \,\cup\, \neg\mathcal{B}_F)}$ that associates a consistent set of literals (beliefs) to each*

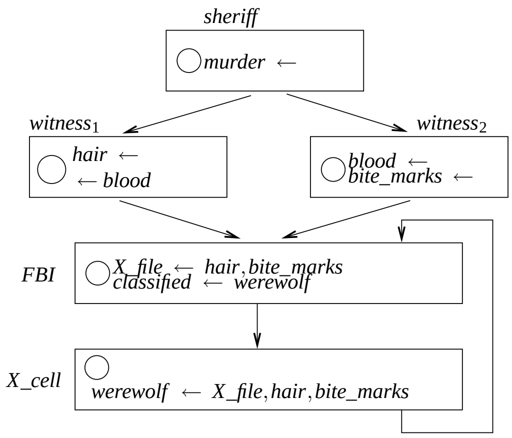


*Figure 3.* The werewolf-killing of Example 9

*agent.*
*Given an interpretation $I$, the* ***inputs*** *and* ***outputs*** *of each agent are defined by $In_I(a) = cons(\cup_{(b,a)\in C} I(b))$ and $Out_I(a) = I(a)$, respectively, where $cons(X) = X^+ \setminus (X^+ \cap X^-)$, i.e. the maximal positive consistent part of $X$.*

Thus, an agent sends its full set of beliefs over all outgoing communication channels. On the other hand, an agent receives as input, the beliefs of all agents connected to its incoming channels. If two agents send conflicting information to a receiving agent, the conflicts are removed.

EXAMPLE 10. *Consider the Werewolf LPAS $F$ of Example 9. We define two interpretations $I_1$ and $I_2$ of $F$.*

$$
\begin{aligned}
&I_1(sheriff) = \{murder\}\\
&I_1(witness_1) = \{hair, \neg blood\}\\
&I_1(witness_2) = \{blood, \neg bite\_marks\}\\
&I_1(FBI) = \{hair, bite\_marks\}\\
&I_1(X\_cell) = \{werewolf\}
\end{aligned}
$$

$$
\begin{aligned}
&I_2(sheriff) = \{murder\}\\
&I_2(witness_1) = \{murder, hair, \neg blood\}\\
&I_2(witness_2) = \{murder, blood, \neg bite\_marks\}\\
&I_2(FBI) = \{murder, hair, bite\_marks, werewolf.X\_file, classified\}\\
&I_2(X\_cell) = \{murder, hair, bite\_marks, werewolf, X\_file, classified\}
\end{aligned}
$$

*The input of agent FBI w.r.t. $I_1$ equals $In_{I_1}(FBI) = \{hair\}$. The output given by the witness$_1$-agent w.r.t. $I_2$ is $Out_{I_2}(witness_1) = \{murder, hair, \neg blood\}$.*

An agent reasons on the basis of positive information that is received from other agents (its input) and its own program that may be used to draw further

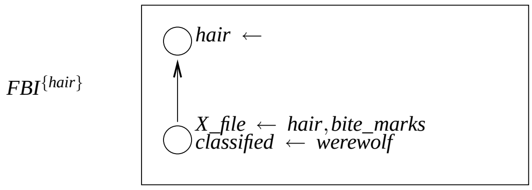


*Figure 4.* The updated version of the *FBI* from Example 11

conclusions, possibly overriding incoming information. Hence, agents attach a higher preference to their own rules rather than to suggestions coming from outside. This can be conveniently modeled by extending an agent's ordered program with an extra "top" component containing the information gathered from its colleagues. This way, the OCLP semantics will automatically allow for defeat of incoming information that does not fit an agent's own program.

DEFINITION 7. *Let $F = \langle \mathbf{A}, C \rangle$ be a LPAS. The* ***updated version*** *of an agent $a \in \mathbf{A}$, with program $F_a = \langle \mathcal{C}_a, \prec_a \rangle$, w.r.t. a set of atoms $U \subseteq \mathcal{B}_F$, denoted $a^U$, is defined by $a^U = \langle \mathcal{C}_a \cup \{c_U\}, \prec_a \cup \{c < c_U \mid c \in \mathcal{C}_a\} \rangle$ with $c_U = \{l \leftarrow \mid l \in U\}$.*

EXAMPLE 11. *The updated version of the FBI-agent w.r.t. $\{hair\}$ is shown in Figure 4.*

For an interpretation to be a model, it suffices that each agent produces a local model (output) that is consistent with its input.

DEFINITION 8. *Let $F = \langle \mathbf{A}, C \rangle$ be a LPAS. An interpretation $I$ of $F$ is a* ***model*** *of $F$ iff $\forall a \in \mathbf{A} \cdot Out_I(a)$ is an answer set of $a^{In_I(a)}$.*

EXAMPLE 12. *Reconsider the Werewolf LPAS of Example 9 and its interpretations $I_1$ and $I_2$ from Example 10. Given the updated version of agent FBI w.r.t. $In_{I_1}(FBI)$ shown in Figure 4, it is easy to see that $I_1$ is not a model. The interpretation $I_2$ on the other hand, is a model.*

For systems without cycles the above model semantics will generate rational solutions for the represented decision-problems. The next example demonstrates that systems that do have cycles may have models that contain too much information, because assumptions made by one agent may become justified by another agent.

EXAMPLE 13. *Two children have been listening to a scary story about vampires and zombies. Suddenly, they think something moved in the room and they start fantasizing about the story they just heard. They come up with the description presented as the LPAS in Figure 5. This system has three models with:*

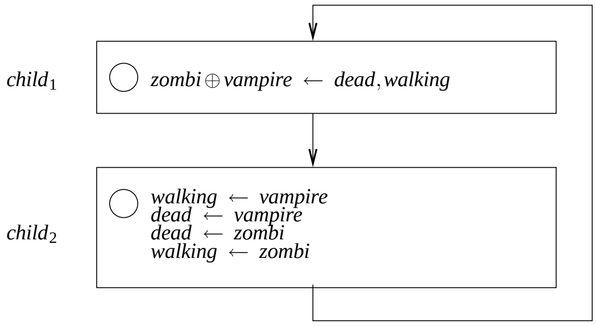


*Figure 5.* The imagination LPAS of Example 13

$M_1(child_1) = M_1(child_2) = \{\neg vampire, \neg zombie, \neg walking, \neg dead\}$,
$M_2(child_1) = M_2(child_2) = \{zombie, walking, dead, \neg vampire\}$,
$M_3(child_1) = M_3(child_2) = \{vampire, walking, dead, \neg zombie\}$ .
*The last two models are not realistic, since the children are just given a description and literals such as zombie are not justified.*

To avoid such self-sustaining propagation of assumptions, we will demand that a model be the result of a fixpoint procedure which mimics the evolution of the belief set of the agents over time.

DEFINITION 9. *Let $F = \langle \mathbf{A}, C \rangle$ be a LPAS. A sequence $I_0, \ldots, I_n$ of interpretations is an* ***evolution*** *of $F$ if for any $i \geq 0$, $a \in \mathbf{A}$, $I_{i+1}(a)$ is an answer set of $a^{In_{I_i}(a)}$.*
*An interpretation $I$ is an* ***evolutionary fixpoint*** *of $F$ w.r.t. an interpretation $I_0$ iff there exists an evolution $I_0, \ldots$ and an integer $i \in \mathbb{N}$ such that $I_j = I_i = I$ for all $j > i$. An* ***answer set*** *of $F$ is an evolutionary fixpoint of $F$ w.r.t. the empty interpretation $I_\emptyset$ (which associates the empty set with each agent).*

Thus, in an evolution, the agents evolve as more information becomes available: at each phase of the evolution, an agent updates her program to reflect input from the last phase and computes a new set of beliefs. An evolution thus corresponds to the way decision-makers try to get a feeling about the other participants. The process of reaching a fixpoint boils down to trying to get an answer to the question "if I do this, how would the other agents react", while trying to establish a stable compromise. Note that the notion of evolution is nondeterministic since an agent may have several local models. For a fixpoint, it suffices that each agent can maintain the same set of beliefs as in the previous stage.

Not every LPAS has an evolutionary fixpoint. Consider e.g. a system describing two children that have to make a choice from two objects *a* and *b*. The first child always wants the object chosen by the other child, while the second child is happy to take whatever the other child does not want. Formally, this can be represented as an LPAS with two interconnected agents $A_1$ (the difficult child) and $A_2$, both being single component OCLP's with rules $\{a_1 \leftarrow a_2\ b_1 \leftarrow b_2\ a_1 \oplus b_1 \leftarrow\}$ for $A_1$ and $\{a_2 \leftarrow b_1\ b_2 \leftarrow a_1\ a_2 \oplus b_2 \leftarrow\}$ for $A_2$, where $a_i$ ($b_i$) stands for "agent $i$ claims object $a$ ($b$)". Clearly, this system alternates between states, without ever reaching a fixpoint.

However, the mechanism of evolutionary fixpoints does allow agents to agree to disagree, since it is not required that the stable models, corresponding to the fixpoint, of each agent in the system should be equal.

EXAMPLE 14. *Consider the Werewolf LPAS of Example 9. The interpretation $I_2$ described in Example 10 is an answer set of the LPAS. The corresponding evolution*[7] *looks like:*

$$\begin{array}{lcl}
V_1(\mathit{sheriff}) & = & \{\mathit{murder}\} \\
V_1(\mathit{witness}_1) & = & \{\mathit{hair}\} \\
V_1(\mathit{witness}_2) & = & \{\mathit{blood}, \mathit{bite_marks}\} \\
V_1(\mathit{FBI}) & = & \emptyset \\
V_1(\mathit{X_cell}) & = & \emptyset \\
\\
V_2(\mathit{sheriff}) & = & \{\mathit{murder}\} \\
V_2(\mathit{witness}_1) & = & \{\mathit{murder}, \mathit{bite_marks}\} \\
V_2(\mathit{witness}_2) & = & \{\mathit{murder}, \mathit{blood}, \mathit{bite_marks}\} \\
V_2(\mathit{FBI}) & = & \{\mathit{X_file}, \mathit{bite_marks}, \mathit{hair}\} \\
V_2(\mathit{X_cell}) & = & \emptyset \\
\\
V_3(\mathit{sheriff}) & = & \{\mathit{murder}\} \\
V_3(\mathit{witness}_1) & = & \{\mathit{murder}, \mathit{bite_marks}\} \\
V_3(\mathit{witness}_2) & = & \{\mathit{murder}, \mathit{blood}, \mathit{bite_marks}\} \\
V_3(\mathit{FBI}) & = & \{\mathit{murder}, \mathit{X_file}, \mathit{bite_marks}, \mathit{hair}\} \\
V_3(\mathit{X_cell}) & = & \{\mathit{X_file}, \mathit{bite_marks}, \mathit{hair}, \mathit{werewolf}\} \\
\\
V_4(\mathit{sheriff}) & = & \{\mathit{murder}\} \\
V_4(\mathit{witness}_1) & = & \{\mathit{murder}, \mathit{bite_marks}\} \\
V_4(\mathit{witness}_2) & = & \{\mathit{murder}, \mathit{blood}, \mathit{bite_marks}\} \\
V_4(\mathit{FBI}) & = & \{\mathit{murder}, \mathit{X_file}, \mathit{bite_marks}, \mathit{hair}, \mathit{werewolf}, \mathit{classified}\} \\
V_4(\mathit{X_cell}) & = & \{\mathit{murder}, \mathit{X_file}, \mathit{hair}, \mathit{bite_marks}, \mathit{werewolf}\} \\
\\
V_5(\mathit{sheriff}) & = & \{\mathit{murder}\} \\
V_5(\mathit{witness}_1) & = & \{\mathit{murder}, \mathit{bite_marks}\} \\
V_5(\mathit{witness}_2) & = & \{\mathit{murder}, \mathit{blood}, \mathit{bite_marks}\} \\
V_5(\mathit{FBI}) & = & \{\mathit{murder}, \mathit{X_file}, \mathit{bite_marks}, \mathit{hair}, \mathit{werewolf}, \mathit{classified}\} \\
V_5(\mathit{X_cell}) & = & \{\mathit{murder}, \mathit{X_file}, \mathit{hair}, \mathit{bite_marks}, \mathit{werewolf}, \mathit{classified}\} \\
\\
V_6 & = & V_5
\end{array}$$

[7] For brevity, we left out negative information.

*The vampire-zombie LPAS of Example 13 has one answer set, namely:* $I(child_1) = I(child_2) = \{\neg zombie, \neg vampire, \neg walking, \neg dead\}$ .

THEOREM 3. *Let* $F = \langle \mathbf{A}, C \rangle$ *be a LPAS. An interpretation* $I$ *is a model of* $F$ *iff it is an evolutionary fixpoint of* $F$ *w.r.t.* $I$.

COROLLARY 1. *Let* $F = \langle \mathbf{A}, C \rangle$ *be a LPAS. Every answer set of* $F$ *is a model of* $F$.

The reverse of the above corollary does not hold in general. A counter example is given in Example 13. However, for acyclic LPAS a one-to-one mapping does exist.

THEOREM 4. *Let* $F = \langle \mathbf{A}, C \rangle$ *be a LPAS without cycles. An interpretation* $M$ *is an answer set iff* $I$ *is a model of* $F$.

## 5. LPAS and Game Theory

In this section we demonstrate that extensive games with perfect information have a natural formalization as logic programming agent systems. The equilibria of such games can be obtained as the answers sets of the system, where each agent represents a player, and the evolution mimics the mechanism players can use in order to come to a decision. The following section contains a brief overview of the relevant game theory background.

### 5.1. Game Theory and Extensive Games with Perfect Information

Game Theory is a growing field of science with roots in mathematics, economics and philosophy. It provides analytical tools designed to improve the understanding of the phenomena observed when decision-makers interact. Modeling such interactions always starts from the assumption that the decision-makers are rational, i.e. the actions they undertake serve a well-defined objective, and that they take into account their knowledge or expectations of the other decision-makers' behavior (e.g. the players of the game act strategically). This way, game theoretic models can anticipate the outcomes of complex interactions with multiple players/actors.

The first publications in the area date back to 1944 ([26]), when game theory was used to examine complex economic behavior. The models of game theory are highly abstract mathematical representations of classes of real-life situations. Their abstractness allows them to be used for the study of a wide spectrum of phenomena. In computer science, game theory has been helpful in obtaining a large number of theoretical results. One of the examples of this is the full abstractness of lazy lambda calculus ([1]). Game theory also contributes to more practical areas of computer science. E.g. the use of evolutionary stable strategies for resource allocation in modern network management improves the utilization significantly [21].

In this paper we restrict our attention to extensive games with perfect information. An extensive game is a detailed description of a sequential structure representing the decision problems encountered by agents (called *players*) in strategic decision making. Players have a preference for certain outcomes over others. This preference is often represented as a payoff function. Each player associates a natural number to each outcome. The larger the number the more preferred is the outcome. The agents in the game are informed of all events that occurred previously. Therefore, they can decide upon their action(s) using information about the actions which have already taken place. This is done by means of passing *histories* of previous actions to the deciding agent. *Terminal histories* are obtained when all the agents/players have made their decision(s).

A good example of an extensive game with perfect information could be a court case where the lawyers of both parties try to win the case. Each party will try to be prepared for every possible move of the other party. However, court cases can be unpredictable; for example the opposing party might come up with some unexpected witness. In such a case the other party will change its course of action. The sequential structure of the game is set by the court system that alternates between the defense and prosecution. Since both parties will closely monitor each other's actions, we can regard this as an extensive game with perfect information.

DEFINITION 10. *An **extensive game with perfect information** is a tuple* $\langle N, H, P, (u_i)_{i \in N} \rangle$ *with the following components:*

- *A set N of players, we assume that* $N = \{1 \ldots n\}$ *for some* $n \in \mathbb{N}$.

- *A prefix-closed*[8] *set H of finite*[9] *sequences. Each element of H is called a **history**; each component of a history is an **action** chosen by a player. A history h is **terminal** if* $\nexists a_{k+1} \cdot (h, a_{k+1}) \in H$. *We use Z to denote the set of terminal histories.*

- *A function P that assigns to each nonterminal history from* $H \setminus Z$ *a member of* $N$. *(P is the **player function**,* $P(h)$ *being the player making a decision after history h).*

[8] A set $X$ of sequences is prefix-closed if $\forall s \in X, \forall x \cdot (x, y) = s \cdot x \in X$.

[9] We restrict ourselves to so-called games with a finite horizon, i.e. games for which all histories are finite.

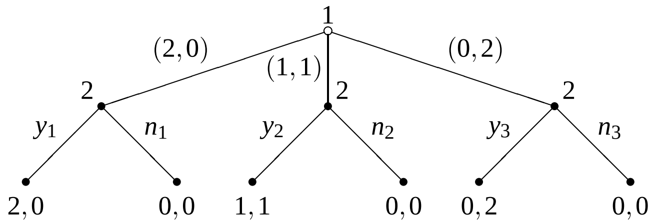


*Figure 6.* The Sharing-an-Object game of Example 15.

- *For each player $i \in N$, a* ***payoff function*** $u_i : Z \to \mathbb{N}$ *for expressing their personal preference over the various terminal histories.*

If $h$ is a history of length $k$ then we denote by $(h,a)$ the history of length $k+1$ consisting of $h$ followed by $a$. Playing the game works as follows: after any nonterminal history $h$ player $P(h)$ chooses an action from the set

$$A(h) = \{a \mid (h,a) \in H\} \ .$$

The empty history is the starting point of the game; we sometimes will refer to it as the *initial* history. At this point player $P(\emptyset)$ chooses a member of $A(\emptyset)$. For each possible choice $a^0$ from this set, player $P(a^0)$ subsequently chooses a member of the set $A(a^0)$; this choice determines the next player's move, and so on. A history after which no more choices have to be made is terminal.

EXAMPLE 15. *Two people use the following procedure to share two desirable identical objects. One of them proposes an allocation, which the other either accepts or rejects. In the event of rejection, neither person receives either of the objects.*
*An extensive game with perfect information that models the individuals' predicament is $\langle N, H, P, (u_i)_{i \in N} \rangle$ where*

- $N = \{1,2\}$ *;*

- *$H$ consists of ten histories* $\emptyset, (2,0), (1,1), (0,2), ((2,0),y_1), ((2,0),n_1), ((1,1),y_2), (1,1),n_2), ((0,2),y_3), ((0,2),n_3)$ *;*

- $P(\emptyset) = 1$ *and* $P(h) = 2$ *for the non-terminal histories* $(2,0), (1,1), (0,2)$ *;*

- $u_1(((2,0),y_1)) = 2$, $u_1(((1,1),y_2)) = 1$, $u_1(((0,2),y_3)) = u_1(((2,0),n_1)) = u_1(((1,1),n_2)) = u_1(((0,2),n_2)) = 0$ *and* $u_2(((0,2),y_3)) = 2$, $u_2(((1,1),y_2)) = 1$, $u_2(((2,0),y_1)) = u_2(((2,0),n_1)) = u_2(((1,1),n_2)) = u_2(((0,2),n_3)) = 0$ .

*A more convenient representation of this game is shown in Fig. 6. The small circle at the top represents the initial history $\emptyset$. The 1 above this circle indicates that $P(\emptyset) = 1$ (player 1 makes the opening move). The three lines that*

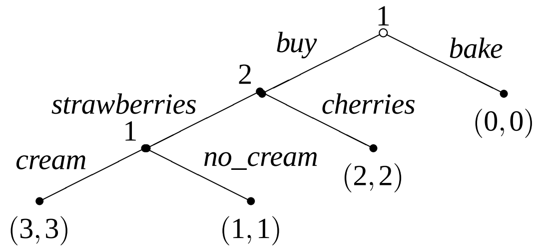


*Figure 7.* The Cake-game of Example 16.

*emanate from the circle correspond to the three members of $A(\emptyset)$ (the possible actions of player 1 at the initial history); the labels beside these line segments are the names of the actions. The rest of the tree is constructed in a similar way. Each path in the tree starting at the root represents a history while a node is a choice point for the player whose turn it is after the history formed by the path from the root to this node. The numbers next to nodes represent the players while the labels of the arcs represent actions. The numbers below a terminal history are the payoffs representing the players' preferences (The first number is the payoff of the first player and the second number is the payoff of the second player).*

A strategy of a player in an extensive game is a plan that specifies the actions chosen by the player when it is her turn to move.

DEFINITION 11. *Let $\langle N, H, P, (u_i)_{i \in N} \rangle$ be a finite extensive game with perfect information. A **strategy** for a player $i \in N$ is a function that assigns an action of $A(h)$ to each non-terminal history $h \in (H \setminus Z)$ for which $P(h) = i$ .*

*A **strategy profile** $s$ is a set containing a strategy for each player $i \in N$, i.e. $s = (s_i)_{i \in N}$ .*

*The **outcome** $O(s)$ for a strategy profile $s$ is defined as the terminal history which is reached when each player $i \in N$ follows the precepts of $s_i$. That is, $O(s)$ is the history $(a_1, \ldots, a_k) \in Z$ such that, for $0 \leq l < k$,*
$s_{P((a_1,\ldots,a_l))}((a_1, \ldots, a_l)) = a_{l+1}$ .

Notice that a strategy of a player $i$ in an extensive game with perfect information does not depend on the payoff function. Looking at the graphical representation of a game, one could say that a strategy solely depends on the structure of the tree, without taking the contents of the leafs into account.

The following example illustrates that a strategy for a player specifies an action for every history after which it is her turn to move, even for histories that, if the strategy is followed are never reached.

EXAMPLE 16. *The game depicted in Figure 7 models the following situation: two ladies have decided that they want fruit cake for dessert. There are* *two possibilities: they either bake a cake or they buy one. At the bakery shop one can choose between strawberry and cherry cake. For strawberry cake there is the possibility to have whipped cream on top. They agree that the first lady will decide on how to get the cake and, if necessary, whether a topping is wanted or not. The second lady will be picking the type of fruit cake. In this game, the first lady has four strategies* $\{bake, cream\}$, $\{bake, no\_cream\}$, $\{buy, cream\}$ *and* $\{buy, no\_cream\}$. *This means that her strategy even specifies an action for the history* $(buy, strawberries)$ *when she decided to go for the action bake in the first place. In this sense, a strategy differs from what we would consider to be a plan of action. Consider now the strategy profile* $\{\{buy, cream\}, cherries\}$. *This profile produces the outcome* $(buy, cherries)$ *which yields a payoff 2 to the first lady and 1 to the second.*

Playing a game $\langle N, H, P, (u_i)_{i\in N}\rangle$ consists of each player $i \in N$ selecting a single strategy. Since players are thought to be rational, it is assumed that a player will select a strategy that leads to some "preferred" profile and corresponding outcome. The problem, of course, is that a player needs to make a decision not knowing precisely what the other players will do.

EXAMPLE 17. *Reconsider the Cake-game from Example 16 and the strategy profile* $s = \{\{buy, cream\}, \{cherries\}\}$. *If the second lady would have known that the first one was thinking of having cream on top of the cake, she would have chosen strawberries instead of the cherries. So s cannot be considered as a good solution to the game. On the other hand, with the strategy* $\{\{buy, cream\},$
$\{strawberries\}\}$, *no lady can benefit from making another decision.*

DEFINITION 12. *A* ***Nash Equilibrium*** *of an extensive game with perfect information* $\langle N, H, P, (u_i)_{i\in N}\rangle$ *is a strategy profile* $s^*$ *such that for every player* $i \in N$ *we have*[10] $u_i(O((s^*_{-i}, s^*_i))) \geq u_i(O((s^*_{-i}, s_i)))$ *for every strategy* $s_i$ *of* $i$ .

Notice that this first solution concept for an extensive game with perfect information ignores the sequential structure of the game; it treats the strategies as choices that are made once and for all, before the actual game starts. Intuitively, a strategy profile $s^*$ is a Nash equilibrium if no player can unilaterally improve upon her choice. In other words, given the other players' actions $s^*_{-i}$, $s^*_i$ is the best player $i$ can do[11].

EXAMPLE 18. *The game of Example 15 has nine Nash equilibria:*
$\{\{(2,0)\}, \{y_1, y_2, y_3,\}\}$, $\{\{(2,0)\}, \{y_1, y_2, n_3\}\}$, $\{\{(2,0)\}, \{y_1, n_2, y_3\}\}$,

[10] $(s_{-i}, s_i)$ is the abbreviation for the strategy profile $s'$ which is such that $s_i = s'_i$ and $s_j = s'_j$ for all $j \in N$ and $j \neq i$.

[11] Note that the strategies of the other players are not actually known to $i$, as the choice for a strategy has been made before the play starts. As stated before, no advantage is drawn from the sequential structure.

$\{\{(2,0)\},\{y_1,n_2,n_3\}\}$, $\{\{(1,1)\},\{n_1,y_2,y_3\}\}$, $\{\{(1,1)\},\{n_1,y_2,n_3\}\}$, $\{\{(0,2)\},\{n_1,n_2,y_3\}\}$, $\{\{(2,0)\},\{n_1,n_2,y_3\}\}$, $\{\{(2,0)\},\{n_1,n_2,n_3\}\}$ .

EXAMPLE 19. *The game of Example 16 has two Nash equilibria:* $\{\{buy, cream\},\{strawberries\}\}$ *and* $\{\{buy, no\_cream\},\{cherries\}\}$ .

Although the Nash equilibria for an extensive game with perfect information have an intuitive definition, they have, in some situations, undesirable properties due to not exploiting the sequential structure of the game. These undesirable properties are illustrated by the following examples.

EXAMPLE 20. *In Example 19 we have seen that the Cake game of Example 16 has two Nash equilibria of which* $\{\{buy, no\_cream\},\{cherries\}\}$ *is one. This strategy profile is unintuitive since it is sustained by the threat that the first lady would opt for no_cream when strawberries are chosen as topping. However, this would never happen (payoff 1) since the first lady prefers cream in this situation (payoff 3).*

Unintuitive equilibria as in the above example may appear because the strategies contributing to a Nash equilibrium are chosen once and for all at the start of game. Players are not allowed to change them during the game. Furthermore, these strategies just have to be optimal as far as the outcome is concerned. The definition does not specify anything about the choices during the game.

These shortcomings can be eliminated by considering subgame perfect equilibria, for which a strategy is required to be optimal after each history, i.e. at each stage of the game.

DEFINITION 13. *A **subgame** of the extensive game* $\Gamma = \langle N, H, P, (u_i)_{i\in N}\rangle$ *that follows the history h is the extensive game* $\Gamma(h) = \langle N, H|_h, P|_h, (u_{h,i})_{i\in N}\rangle$, *where* $H|_h$ *is the set of sequences* $h'$ *of actions for which* $(h,h') \in H$, $P|_h$ *is defined by* $P|_h(h') = P((h,h'))$ *for each* $h' \in H|_h$, *and* $u_{h,i}(h') = u_i((h,h'))$.

A *subgame perfect equilibrium* requires that the actions prescribed by each player's strategy are optimal, given the other player's strategies, and this after *every* history. Given a strategy $s_i$ of player $i$ and a history $h$ in the extensive game $\Gamma$, $s_i|_h$ denotes the strategy that $s_i$ induces in the subgame $\Gamma(h)$ (i.e. $s_i|_h(h') = s_i((h,h'))$). We will use $O_h$ to denote the outcome function of $\Gamma(h)$.

DEFINITION 14. *A **subgame perfect equilibrium** of an extensive game with perfect information* $\Gamma = \langle N, H, P, (u_i)_{i\in N}\rangle$ *is a strategy profile* $s^*$ *such that for every player* $i \in N$ *and every non-terminal history* $h \in H \setminus Z$ *for which* $P(h) = i$ *we have:* $u_{h,i}(O_h(s^*_{-i}|_h, s^*_i|_h)) \geq u_{h,i}(O_h((s^*_{-i}|_h, s_i|_h))$ , *for every strategy* $s_i$ *of player i in the subgame* $\Gamma(h)$.

Equivalently, we can say that a subgame perfect equilibrium needs to be a Nash equilibrium in every subgame. Because subgames start from the bottom of the tree and then working themselves up to the top, this mechanism is often called backward induction. Every subgame can be seen as that part of the game that is still uncertain, i.e. the future. The part above this subgame is common knowledge to all players, since we are dealing with an extensive game with perfect information. Thus, these decisions are already something from the past. So the player that has to make a decision at the bottom of the game tree does not have to make any assumption on the behavior of the other players. Consequently, she only has to worry about her own benefit. The player who made the previous decision can anticipate this choice and can take it into account. This process works its way back to the top. Another way of looking at this is that the players revise their strategy after every choice made by any player.

Subgame perfect equilibria eliminate Nash equilibria in which the players' threats are not credible.

EXAMPLE 21. *The Cake-game of Example 16 admits only one subgame perfect equilibrium:* $\{\{buy, cream\}, \{strawberries\}\}$.
*The Object-game from Example 15 has only two subgame perfect equilibria, namely* $\{\{(2,0), \{y_1y_2y_3\}\}$ *and* $\{\{(1,1)\}, \{n_1y_2y_3\}\}$. *The other seven Nash equilibria are based on implausible threats.*

Since subgame perfect equilibria are Nash equilibria for every subgame, the following theorem should not be surprising.

THEOREM 5 ([22]). *Let* $\Gamma = \langle N, H, P, (u_i)_{i \in N} \rangle$ *be an extensive game with perfect information. Then, every subgame perfect equilibrium for* $\Gamma$ *is also a Nash equilibrium.*

## 5.2. Playing Games

We demonstrate that extensive games with perfect information have a natural formulation as multi-agent systems with a particularly simple information-flow structure between the agents. We introduce the mapping in three steps, each with more emphasis on the players and their interactions. For our mappings, we assume that an action can only appear once[12]. This is not really a restriction, since one can simply use different names for these actions since they are not related. This will just have an effect on the syntax, and not on the semantics, of the game.

[12] Formally, for any two histories $h_1, h_2 \in H : A(h_1) \cap A(h_2) = \emptyset$.

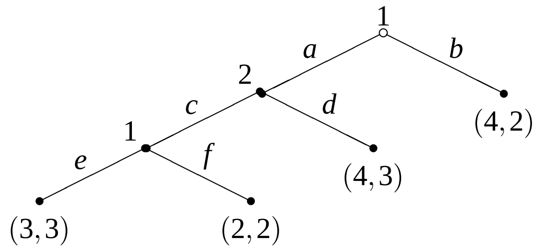


*Figure 8.* The game of Example 22

*Phase 1 – games as OCLP's*
Extensive games can be represented as ordered choice logic program in such a way that, depending on the transformation, either the Nash or subgame perfect equilibria can be retrieved as the answer sets of the program [11].

EXAMPLE 22. *Consider the extensive game depicted in Figure 8. This game has six Nash equilibria:* $\{\{b,e\},\{c\}\}$, $\{\{b,f\},\{c\}\}$, $\{\{b,e\},\{d\}\}$,$\{\{b,f\},\{d\}\}$, $\{\{a,e\},\{d\}\}$,$\{\{a,f\},\{d\}\}$ *. Three of these Nash equilibria are also subgame perfect equilibria:* $\{\{b,e\},\{c\}\}$,$\{\{b,e\},\{d\}\}$,$\{\{a,e\},\{d\}\}$ *.*

The following transformations will be used to retrieve the Nash and subgame perfect equilibria of a game as the answer sets of the corresponding OCLPs $P_N$ and $P_S$, respectively.

DEFINITION 15. *Let* $G = \langle N, H, P, (u_i)_{i\in N}\rangle$ *be a finite extensive game with perfect information. Then, the program* $P_N$ *is defined as follows:*

1. $\mathcal{C} = \{C^t\} \cup \{C_u \mid \exists i \in N, h \in Z \cdot u = u_i(h)\}$ ;
2. $C^t \prec C_u$ *for all* $C_u \in \mathcal{C}$ ;
3. $\forall C_u, C_w \in \mathcal{C} \cdot C_u \prec C_w$ *iff* $u > w$ ;
4. $\forall h \in (H \setminus Z) \cdot (A(h) \leftarrow\ ) \in C^t$ ;
5. $\forall h = h_1 a h_2 \in Z \cdot (a \leftarrow B) \in C_u$ *with* $B = \{b \in [h]^{13} \mid h = h_3 b h_4, P(h_3) \neq P(h_1)\}$ *and* $u = u_{P(h_1)}(h)$ .

*Furthermore, the program* $P_S$ *is defined like* $P_N$*, except that item 5 is replaced with 5':*

5'. $\forall h = h_1 a h_2 \in Z \cdot (a \leftarrow B) \in C_u$ *with* $B = \{b \in [h_2] \mid h = h_3 b h_4, P(h_3) \neq P(h_1)\}$ *and* $u = u_{P(h_1)}(h)$ .

[13] We use $[h]$ to denote the set of actions appearing in a sequence $h$.

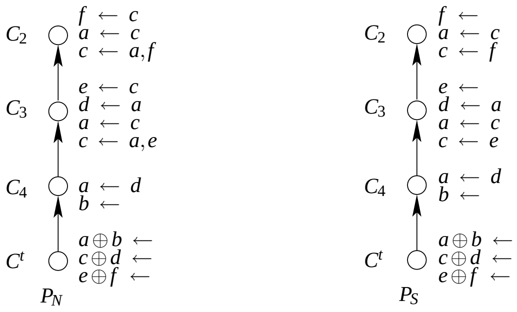


*Figure 9.* The OCLP's of Example 22

Intuitively, $P_N$ contains a component $C^t$ containing all the decisions that need to be considered and a component $C_u$ for each payoff $u$. The order among the components follows the expected payoff (higher payoffs correspond to more specific components) with the decision component at the bottom of the hierarchy (the most specific component). Since Nash equilibria do not take into account the sequential structure of the game, players have to decide upon their strategy before starting the game, leaving them, for each decision, to reason about both past and future. This is reflected in the rules of $P_N$ (item 5): each rule in a payoff component is made out of a terminal history (path from top to bottom in the tree) where the head represents the action taken when considering the past and future created by the other players according to this history. The component of the rule corresponds to the payoff the deciding player would receive in case the history was actually followed.

The construction of $P_S$ is quite similar to the one for $P_N$. The only difference between the two is in the history-dependent rules of $P_S$ (item 5'): since subgame perfect equilibria take the sequential structure into account, players no longer need to reason about what happened before they have to make their decision. They can focus solely on the future.

EXAMPLE 23. *Figure 9 depicts the corresponding OCLPs $P_N$ and $P_S$, constructed according to Definition 15 from the game of Example 22. Notice that the answer sets of $P_N$ and $P_S$ coincide exactly with respectively the Nash and subgame perfect equilibria of the game.*

The following theorem demonstrates that OCLPs can indeed be used to retrieve the equilibria of extensive games.

THEOREM 6 ([11]). *Let $G = \langle N, H, P, (u_i)_{i \in N} \rangle$ be a finite extensive game with perfect information and let $P_N$ and $P_S$ be its corresponding OCLPs according to Definition 15. Then, $s^*$ is a Nash equilibrium (resp. subgame perfect equilibrium) for $G$ iff $s^*$ is a answer set of $P_N$ (resp. $P_S$).*

*Phase 2 – players as sets of components*
The above transformations ignore the natural player structure of a game. It is however fairly simple to adapt $P_N$ and $P_S$ such that each player is associated with her own set of components. Instead of having a payoff component for every payoff in the game, we introduce payoff components corresponding to the distinct payoffs of each player (e.g. if two players $i$ and $j$ have a payoff 0 then we now have two components $C_0^i$ and $C_0^j$, instead of the single component $C_0$). Rules made out of a terminal history are now put in the component corresponding to the player taking the associated decision and her perceived payoff.

The order among the components is established, first among components of the same player, according to their payoff (lower payoff is more general) and, secondly, according to the position of the players in $N$.

DEFINITION 16. *Let $\langle N,H,P,(u_i)_{i\in N}\rangle$ be a finite extensive game with perfect information. The program $P_N^p$ is defined by*

1. $\mathcal{C} = \{C_u^i \mid \exists h \in Z \cdot u = u_i(h) \wedge i \in N\}$ ;
2. $\forall C_u^i, C_w^j \in \mathcal{C} \cdot C_u^i \prec C_w^j$ *iff* $i > j$ ;
3. $\forall C_u^i, C_w^i \in \mathcal{C} \cdot C_u^i \prec C_w^i$ *iff* $u > w$ ;
4. $\forall h \in (H \setminus Z) \cdot (A(h) \leftarrow\ ) \in C_w^i, P(h) = i, \forall C_n^i, n \neq w \cdot C_w^i \prec C_n^i$ ;
5. $\forall h = h_1 a h_2 \in Z \cdot (a \leftarrow B) \in C_u^i$ *with* $P(h_1) = i$,
   $B = \{b \in [h] \mid h = h_3 b h_4,\ P(h_3) \neq i\}$ *and* $u = u_{P(h_1)}(h)$ .

*The definition of $P_S^p$ is similar, it suffices to replace condition 5 by:*

5′. $\forall h = h_1 a h_2 \in Z \cdot (a \leftarrow B) \in C_u^i$ *with* $P(h_1) = i$, $B = \{b \in [h_2] \mid h = h_3 b h_4, P(h_3) \neq i\}$ *and* $u = u_{P(h_1)}(h)$ .

EXAMPLE 24. *The programs $P_N^p$ and $P_S^p$ for the game of Example 22 are shown in Figure 10. Notice that the answer sets of $P_N^p$ match the Nash equilibria of the game, while the answer sets of $P_S^p$ are the subgame perfect equilibria.*

THEOREM 7. *Let $\langle N,H,P,(u_i)_{i\in N}\rangle$ be a finite extensive game with perfect information and let $P_N^p$ and $P_S^p$ be its corresponding OCLP's. Then, $s^*$ is a Nash equilibrium (resp. subgame perfect equilibrium) for $\langle N,H,P,(u_i)_{i\in N}\rangle$ iff $s^*$ is an answer set of $P_N^p$ (resp. $P_S^p$).*

*Phase 3 – games as LPASs*
The ordered programs of Definition 16 can be transformed into equivalent cyclic agent systems where there is a one-to-one correspondence between

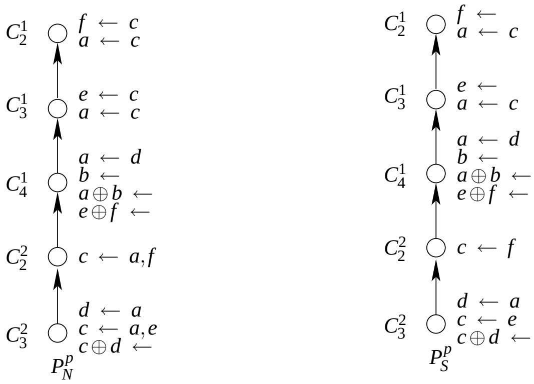


*Figure 10.* OCLP's for Example 24

agents and players. Intuitively, the OCLPs of phase 2 are divided into smaller OCLPs, each describing the reasoning skill of a single player. Each of them will be used to create an agent. Agents are connected in a single directed cycle.

DEFINITION 17. *Let $\langle N, H, P, (u_i)_{i \in N} \rangle$ be a finite extensive game with perfect information. The corresponding Nash LPAS $S^N = \langle \{A^i \mid i \in N\}, C \rangle$ where $A^i = \langle \{\mathcal{C}_{A^i}\}, \prec_i \rangle$ is constructed as follows:*

1. $\mathcal{C}_{A^i} = \{C_u^i \mid \exists h \in Z \cdot u = u_i(h)\}$ ;
2. $\forall C_u^i, C_w^i \in \mathcal{C}_{A^i} \cdot C_u^i \prec_i C_w^i$ *iff* $u > w$ ;
3. $\forall h \in (H \setminus Z), P(h) = i \cdot (A(h) \leftarrow\ ) \in C_w^i, \forall C_n^i, n \neq w \cdot C_w^i \prec_i C_n^i$ ;
4. $\forall h = h_1 a h_2 \in Z, P(h_1) = i \cdot a \leftarrow B \in C_u^i$ *with* $B = \{b \in [h] \mid h = h_3 b h_4,\ P(h_3) \neq i\}$ *and* $u = u_{P(h_1)}(h)$ ,
5. $C(A^i) = \{A^{i+1}\}$ *for* $i \in N, i < \max N$ ,
6. $C(A^{\max N}) = \{A^1\}$ .

*The LPAS $S^S$ can be obtained in a similar way by replacing 4 by:*

$4'$. $\forall h = h_1 a h_2 \in Z, P(h_1) = 1 \cdot (a \leftarrow B) \in C_u^i$ *with* $B = \{b \in [h_2] \mid h = h_3 b h_4, P(h_3) \neq i\}$ *and* $u = u_{P(h_1)}(h)$ .

EXAMPLE 25. *For the game of Example 22, the corresponding LPASs $S^N$ and $S^S$ are displayed in Figure 11. Notice that the answer sets for $S^N$ and $S^S$ match exactly the Nash and subgame perfect equilibria of the game.*

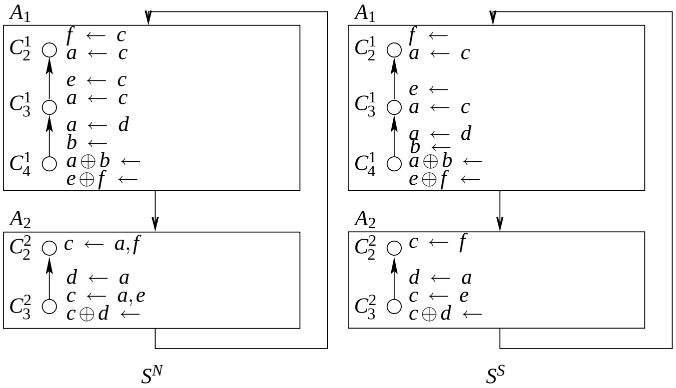


*Figure 11.* The LPASs of Example 25

THEOREM 8. *Let $\langle N, H, P, (u_i)_{i \in N} \rangle$ be a finite extensive game with perfect information and let $S^N$ and $S^S$ be the corresponding LPASs, according to Definition 17. Then, $s^*$ is a Nash equilibrium (resp. subgame perfect equilibrium) for $\langle N, H, P, (u_i)_{i \in N} \rangle$ iff the interpretation $I$ with $I(a) = s^*$ for every $a \in \mathbf{A}$ is an answer set of $S^N$ (resp. $S^S$).*

For the proof of the above theorem, we demonstrate that every evolutionary fixpoint of $I_\emptyset$ can be constructed in $n$ iterations, with $n$ the number of players in the game. Of course, this only happens when players or agents know which actions will lead to an equilibrium state. In practice, it might take more iterations in order to find a fixpoint. Such a fixpoint computation can easily be seen as the players trying to obtain actions belonging to an equilibrium state. At first, she picks an action and sees how the other players respond to this. With this information she can update her actions. This process is carried on until a equilibrium is reached.

## 6. Relationship to Other Approaches

### 6.1. (Extended) Logic Programming

Choice logic programs can represent logic programs [11]. However, the mapping between the stable models/answer sets of both formalisms is not total since an extra condition for the stable models of a choice logic program, namely rationality, is required. A full mapping exists only for the class of positive-acyclic logic programs. Generalizing to OCLP, we can obtain a full one-to-one correspondence, as far as answer sets are concerned, between logic programs and their corresponding OCLPs. The mapping can be further extended to simulate the answer set semantics of extended logic programs.

DEFINITION 18. *Let $P$ be an extended logic program without classical negation. The corresponding OCLP $P_L$ is defined by $\langle \{C,R,N\}, C \prec R \prec N \rangle$ with*

$$
\begin{aligned}
N &= \{\mathrm{not}_a \leftarrow \ \mid a \in \mathcal{B}_P\} \ , \\
R &= \{a \leftarrow B, \mathrm{not}_C \in R \mid r : a \leftarrow B, \neg C \in P\} \cup \\
&\quad \{\mathrm{not}_a \leftarrow B, \mathrm{not}_C \in R \mid r : \neg a \leftarrow B, \neg C \in P\} \cup \\
&\quad \{\ \leftarrow B, \mathrm{not}_C \in R \mid r : \ \leftarrow B, \neg C \in P\} \ , \\
C &= \{a \oplus \mathrm{not}_a \leftarrow a \mid a \in \mathcal{B}_P\} \ ,
\end{aligned}
$$

*where, for $a \in \mathcal{B}_P$, $\mathrm{not}_a$ is a fresh atom representing $\neg a$[14].*

Intuitively, the choice rules in $C$ force a choice between $a$ and $\neg a$ while the rules in $N$ encode "negation by default".

THEOREM 9. *Let $P$ be an extended logic program without classical negation[15]. Then, $M \subseteq \mathcal{B}_P$ is an answer set of $P$ iff $S$ is an answer set of $P_L$ with $S^+ = M^+ \cup \mathrm{not}_{(\mathcal{B}_P \setminus M)}$.*

The proof of this theorem relies on the choice rules $a \oplus \mathrm{not}_a \leftarrow a$ to obtain the one-to-one mapping between the answer sets. The next example demonstrates that this is essential.

EXAMPLE 26. *Consider the very simple logic program $P$: $a \leftarrow a$ . Obviously, we obtain $\emptyset$ as the only answer set of this program. When we apply the transformation of Definition 18, we obtain a OCLP $P_L$ with a single answer set $M$ with $M^+ = \{\mathrm{not}_a\}$. Suppose we would use choice rules with empty body. Then the program would produce two answer set: $M$ and $N$ with $N^+ = \{a\}$. Certainly, $N$ does not correspond to any answer set of $P$.*

## 6.2. PREFERENCES

Various logic (programming) formalisms have been introduced to deal with the notions of preference, order and updates. Ordered choice logic programming uses the intuition of defeating from ordered logic programming (OLP) [17, 18] to select the most favorable alternative of a decision. In fact, every ordered logic program can be transformed into a OCLP such that the answer set semantics reflects the credulous semantics of the OLP.

Dynamic preference in extended logic programs was introduced in [4] in order to obtain a better suited well-founded semantics. Although preferences

[14] For a set $X \in \mathcal{B}_P$, $\mathrm{not}_X = \{\mathrm{not}_a \mid a \in X\}$.

[15] This is not a real restriction: classical negation can easily be replaced using a simple preprocessor.

are called dynamic they are not dynamic in our sense. Instead of defining a preference relation on subsets of rules, preferences are incorporated as rules in the program. Moreover, a stability criterion may come into play to overrule preference information. Another important difference with our approach is the notion of alternatives, as the corresponding notion in [4] is statically defined.

A totally different approach is proposed in [25]. where preferences are defined between atoms. Given these preferences, one can combine them to obtain preferences for sets of atoms. Defining models in the usual way, the preferences are then used to filter out the less preferred models.

[6] proposes disjunctive ordered logic programs which are similar to ordered logic programs [17] where disjunctive rules are permitted. In [5], preference in extended disjunctive logic programming is considered. As far as overriding is concerned the technique corresponds rather well with a skeptical version of the OCLP semantics ([13]), but alternatives are fixed as an atom and its (classical) negation.

To reason about updates of generalized logic programs, extended logic programs without classical negation, [2] introduces dynamic logic programs. A stable model of such a dynamic logic program is a stable model of the generalized program obtained by removing the rejected rules. The definition of a rejected rule corresponds to our definition of a defeated rule when $a$ and $\neg a$ are considered alternatives. It was shown in [2], that the stable model semantics and the answer set semantics coincide for generalized logic programs. In Theorem 9 we have demonstrated that extended logic programs without classical negation can be represented as ordered choice logic programs such that the answer set semantics of the extended logic program can be obtained as the answer set semantics of the OCLP. Because rejecting rules corresponds to defeating rules, it is not hard to see that, with some minor changes, Definition 18 can be used to retrieve the stable models of the dynamic logic program as the stable models of the corresponding OCLP. The only things we need to do are to replace the component $R$ by the $P_i$s of the dynamic logic program $\bigoplus\{P_i : i \in S\}$, replace every occurrence of $\neg a$ by $\mathrm{not}_a$ and add $a \oplus \mathrm{not}_a \leftarrow \mathrm{not}_a$ to $C$ for each $a \in \mathcal{B}_P$.

A similar system is proposed in [15], where sequences are based on extended logic programs, and defeat is restricted to rules with opposing heads. The semantics is obtained by mapping to a single extended logic program containing expanded rules such that defeated rules become blocked in the interpretation of the "flattened" program.

A slightly different version of Definition 18 can be used to map the sequences of programs of [15] to OCLPs.

In [3], preferences are added to the dynamic logic program formalism of [2]. These are used to select the most preferred stable models. Along the same line, [14] proposes logic programs with compiled preferences, where preferences may appear in any part of the rules. For the semantics, [14] maps the program to an extended logic program.

### 6.3. Games

In Section 5 we demonstrated that extensive games with perfect information have a natural formulation as a system of logic programming agents such that equilibria coincide with answer sets. Furthermore, the fixpoint semantics simulates the way players make assumptions about other players' behaviors in order to make a decision.

Besides explaining and retrieving game theoretic phenomena, OCLPs can be used to extend game theory. First of all, OCLPs can be used to represent more complex games since rules may involve extra antecedents and dependencies that cannot easily be represented in games. Perhaps even more important is the ability for a player to take more than one action as demonstrated by the Travel OCLP of the introduction (Example 2). If we just consider the first three components ($P_1$, $P_2$ and $P_3$), we see the representation of a very simple strategic or extensive game with a single player (the person who wants to go on vacation). In this case the equilibrium would be $\{spain\}$ which corresponds to the situation with a smaller budget. When you win the lottery, your budget will be considerably larger and you will be able to afford two vacations instead of one. In game theory this is simply not possible. Every player is forced to take a single action. Another advantage of using (ordered) logic programming for game theory is its ability not only to serve as a test lab for game theory but also as an implementation tool to obtain the equilibria of game.

To the best of our knowledge, little work has been done so far on game theory in the context of logic programming. An important exception is [23] which introduces a formalism called "Independent Choice Logic" (ICL) which uses (acyclic) logic programs to deterministically model the *consequences* of choices made by agents. Since choices are external to the logic program, [23] restricts the programs further to be not only deterministic (i.e. each choice leads to a unique stable model) but also independent in the sense that literals representing alternatives may not influence each other, e.g. they may not appear in the head of rules. ICL is further extended to reconstruct much of classical game theory and other related fields.

### 6.4. Agents

A considerable amount of work has been done in the area of logic programming and agents. Due to space restrictions, we only mention systems designed for game theoretic purposes. [24] investigates methods to prevent agents exploiting game theoretic properties of negotiations: if, e.g., the players in a task oriented domain know that all players follow the game theory route, they can exploit this knowledge by introducing phantom tasks or by hiding tasks to improve their score. [23] incorporates the players of the game directly into a logic programming formalism for strategic games in order to obtain mixed strategy Nash equilibria (where probabilities are introduced by an additional mechanism that is external to the logic program). Here, on the other hand, we are interested in multi-agent systems that are able to represent, in an intuitive way, games such that agents correspond with players and models with the equilibria.

## 7. Conclusions and Directions for Future Research

In this paper we proposed a logic programming agent systems that allows us to represent the communication between decision-makers in order to come to a conclusion. The semantics of such a system relies on the agents to send answer sets to connected agents. Furthermore, we demonstrated that extensive games with perfect information have an elegant and natural formalization as a LPAS. The Nash equilibria and subgame perfect equilibria can be obtained as the answer sets of the corresponding system. In addition, the fixpoint computation reflects the way players reason in order to make a decision belonging to an equilibrium state.

A number of questions re. LPAS remain to be studied: e.g. what are sufficient conditions on the agent programs and/or their communication structure for guaranteeing the existence of an answer set (see Section 4 for an example of a LPAS that has no answer sets)?

Although logic programming has shown itself to be a convenient representational language for decision-problems, we only covered a few aspects of the decision-making process. E.g. we assume that the agents are rational and fair and that none of them has a hidden agenda or would deliberately deceive the other agents. To model such aspects of decision-making, we need to add epistemic primitives to the formalism such that agents can reason about each other in more detail.

An extension along the same line is information hiding by agents. The current definition of LPAS assumes that every agent is willing to share all her information, in the form of an answer set, with the connected agents, i.e. only the reasoning capabilities (the program) are hidden. There are situations in which this may not be desirable, as when consequences of classified information may be made public without revealing the underlying motivation.

By demonstrating that extended logic programs can be represented in OCLP, we obtain a lower complexity bound for OCLP. Current research [8] involves mapping OCLP to logic programs, thus showing that the complexity of OCLP equals that of logic programming. Having such a result implies that one could use dlv [16] and/or smodels [20] to supply an implementation for

OCLP and, consequently, LPAS. Smodels seems to be a particularly suitable candidate for this, as it already contains primitives to express priority and exclusive choice. Because the game OCLPs/LPASs are very specific, it might be possible to incorporate this knowledge in the algorithms in order to obtain the equilibria more efficiently.

## 8. Acknowledgments

The authors would like to take this opportunity to thank the anonymous referees for their comments, which helped improving this paper and provided us with new ideas and insights. This work was partially supported by the European Fifth Framework Programme under the grant IST-2001-37004 (WASP).

## References

1. Abramsky, S. and C.-H. L. Ong: 1993, 'Full Abstraction in the Lazy Lambda Calculus'. *Information and Computation* **105**(2), 159–276.
2. Alferes, J. J., L. J. A., L. M. Pereira, H. Przymusinska, and T. C. Przymusinski: 1998, 'Dynamic Logic Programming'. in [7], pp. 98–111.
3. Alferes, J. J. and L. M. Pereira: 2000, 'Updates plus Preferences'. In: *European Workshop, JELIA 2000*, Vol. 1919 of *Lecture Notes in Artificial Intelligence*. Malaga, Spain, pp. 345–360.
4. Brewka, G.: 1996, 'Well-Founded Semantics for Extended Logic Programs with Dynamic Preferences'. *Journal of Articficial Intelligence Research* **4**, 19–36.
5. Buccafurri, F., W. Faber, and N. Leone: 1999, 'Disjunctive Logic Programs with Inheritance'. In: D. De Schreye (ed.): *International Conference on Logic Programming (ICLP)*. Las Cruces, New Mexico, USA, pp. 79–93.
6. Buccafurri, F., N. Leone, and P. Rullo: 1998, 'Disjunctive Ordered Logic: Semantics and Expressiveness'. in [7], pp. 418–431.
7. Cohn, A. G., L. K. Schubert, and S. C. Shapiro (eds.): 1998, 'Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning'. Trento:, Morgan Kaufmann.
8. De Vos, M., 'Implementing Ordered Choice Logic Programs'. *In preparation*.
9. De Vos, M. and D. Vermeir: 1999a, 'Choice Logic Programs and Nash Equilibria in Strategic Games'. In: J. Flum and M. Rodríguez-Artalejo (eds.): *Computer Science Logic (CSL'99)*, Vol. 1683 of *Lecture Notes in Computer Science*. Madrid, Spain, pp. 266–276.
10. De Vos, M. and D. Vermeir: 1999b, 'On the Role of Negation in Choice Logic Programs'. In: M. Gelfond, N. Leone, and G. Pfeifer (eds.): *Logic Programming and Non-Monotonic Reasoning Conference (LPNMR'99)*, Vol. 1730 of *Lecture Notes in Artificial Intelligence*. El Paso, Texas, USA, pp. 236–246.
11. De Vos, M. and D. Vermeir: 2000, 'A Logic for Modelling Decision Making with Dynamic Preferences'. In: *Proceedings of the Logic in Artificial Intelligence (Jelia2000) workshop*. Malaga, Spain, pp. 391–406.
12. De Vos, M. and D. Vermeir: 2001, 'Logic Programming Agents and Game Theory'. In: *Answer Set Programming: Towards Efficient and Scalable Knowledge Representation and Reasoning*. Stanford (Palo Alto), California, US, pp. 27–33.

13. De Vos, M. and D. Vermeir: 2002, 'Dynamic Decision Making in Logic Programming and Game Theory'. In: *AI2002: Advances in Artificial Intelligence*. pp. 36–47.
14. Delgrande, J., T. Schaub, and H. Tompits: 2000, 'Logic Programs with Compiled Preferences'. In: W. Horn (ed.): *European Conference on Artficial Intelligence*. pp. 392–398.
15. Eiter, T., M. Fink, G. Sabbatini, and H. Tompits: 2002, 'On Properties of update Sequences Based on Causal Rejection'. *Theory and Practice of Logic Programming* **2**(6).
16. Eiter, T., N. Leone, C. Mateis, G. Pfeifer, and F. Scarcello: 1998, 'The KR System `dlv`: Progress Report, Comparisons and Benchmarks'. In: A. G. Cohn, L. Schubert, and S. C. Shapiro (eds.): *KR'98: Principles of Knowledge Representation and Reasoning*. San Francisco, California: Morgan Kaufmann, pp. 406–417.
17. Gabbay, D., E. Laenens, and D. Vermeir: 1991, 'Credulous vs. Sceptical Semantics for Ordered Logic Programs'. In: J. Allen, R. Fikes, and E. Sandewall (eds.): *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*. Cambridge, Mass, pp. 208–217.
18. Laenens, E. and D. Vermeir: 2000, 'A Universal Fixpoint Semantics for Ordered Logic'. *Computers and Artificial Intelligence* **19**(3).
19. Lifschitz, V.: 2002, 'Answer set programming and plan generation'. *Journal of Artificial Intelligence* **138**(1-2), 39–54.
20. Niemelä, I. and P. Simons: 1997, 'Smodels: An implementation of the stable model and well-founded semantics for normal LP'. In: J. Dix, U. Furbach, and A. Nerode (eds.): *Proceedings of the 4th International Conference on Logic Programing and Nonmonotonic Reasoning*, Vol. 1265 of *LNAI*. Berlin, pp. 420–429.
21. Olafsson, S.: 1996, 'Resource Allocation as an Evolving Strategy'. *Evolutionary Computation* **4**(1), 32–54.
22. Osborne, M. J. and A. Rubinstein: 1996, *A Course in Game Theory*. Cambridge, Massachusets, London, Engeland: The MIT Press, third edition.
23. Poole, D.: 1997, 'The Independent Choice Logic for Modelling Multiple Agents Under Uncertainty'. *Artificial Intelligence* **94**(1–2), 7–56.
24. Rosenschein, J. S. and G. Zlotkin: 1994, *Rules of Encounter. Designing Conventions for Automated Negotiation among Computers*. The MIT Press.
25. Sakama, C. and K. Inoue: 1996, 'Representing Priorities in Logic Programs'. In: M. Maher (ed.): *Proceedings of the 1996 Joint International Conference and Symposium on Logic Programming*. Cambridge, pp. 82–96.
26. Von Neumann, J. and O. Morgenstern: 1944, *Theory of Games and Economic Behavior*. New York, US: John Wiley and Sons.


# Appendix

## A. Proofs

### A.1. Ordered Choice Logic Programming

THEOREM 1 *Let $P$ be a choice logic program. $M$ is a stable model for $P$ if $M$ is an answer set of the corresponding OCLP $P_o = \langle \{P\}, \prec \rangle$.*

*Proof.* In order to demonstrate that $M$ is answer set of $P_o$, we need, by Definition 4 to show that $M$ is a minimal model of $P_o^M$.

– $M$ is a model of $P_o^M$. Because $M$ is a stable model of $P$, we have that for every rule $r \in P$ applicable rule holds $|H_r \cap M| = 1$. Because rules are left unchanged in the reduction process, we have that all rules in $P_o^M$ are satisfied.

– $M$ is a minimal model of $P_o^M$. This implies that we need to show that no interpretation $I$ of $P_o^M$ is a model of it. Since $M$ is a stable model of $P$, we know that $M$ is minimal and therefore $I$ cannot be a model of $P$. This means a rule $r$ exists such that $B_r \subseteq I \subseteq M$ while $|H_r \cap I| \neq 1$. Since $M$ is a model of $P$, we must have $|H_r \cap M| = 1$. For us to proof that $I$ is not a model of $P_o^M$, it suffices to demonstrate that $r$ is part of $P_o^M$, which means showing that $r$ is not defeated w.r.t. $M$. When $M$ is concerned, we know that $r$ is applied. Assume that $H_r \cap M = \{a\}$. Suppose know that $r$ would be defeated. According to Definition 3, this implies, among other things, $\exists b \in (\Omega^M_{c(r)}(a) \cap M) \neq \emptyset$ . With Definition 2, this yields $\exists r' \in P \cdot a \oplus b \oplus B \leftarrow C$ such that $C \subseteq M$. Clearly this rule is applicable but unapplied, which is contradiction with $M$ being a model of $P$. This leaves us no other option then to conclude $r$ cannot be defeated and will therefore be in $P_o^M$. Which is enough to state that $I$ could impossibly be a model of $P_o^M$. Therefore, $M$ is a minimal model of $P_o^M$.

THEOREM 2 *Let $M$ be an answer set of an OCLP $P$. Then, $M$ is a model for $P$.*

*Proof.* In order to show this, we need to prove that every rule in $P^\star$ is either non-applicable, applied or defeated w.r.t. $M$. Let us begin with the constraints. Since they do not contain head atoms, they cannot be defeated, implying that they are automatically included in $P_c^M$. Since $M$ is a stable model of $P_c^M$ and thus a model, we have that every constraint is non-applicable in $P_c^M$, which immediately implies that they are also satisfied for $P$.
Let us now consider the rules with head atoms. Let $r$ be such a rule. Assume that $r$ is applicable and not defeated w.r.t. $M$. If one of these conditions is not met, we have that $r$ is satisfied w.r.t. $M$. Since $r$ is not defeated w.r.t. $M$, $r$ must be in $P_c^M$. Since $M$ is model and $r$ is applicable, $r$ has to be applied in $P_c^M$, which implies $|H_r \cap M| = 1$, making $r$ applied in $P$.
Since all rules in $P^\star$ are satisfied, we obtain that $M$ is a model for $P$.

## A.2. Logic Programming Agents

THEOREM 3 *Let $F = \langle \mathbf{A}, C \rangle$ be a LPAS. An interpretation $I$ is a model of $F$ iff it is an evolutionary fixpoint of $F$ w.r.t. $I$.*

*Proof.* Let $I$ be a model of $F$. It is easy to see that the sequence $I, I$ satisfies the criteria for a evolution. So, with Definition 9, $I$ is a evolutionary fixpoint of $F$ w.r.t. $I$.
Let $I$ be a evolutionary fixpoint. With Definition 9, an evolution $I_0, \ldots, I_n = I$ exists such that, for all $a \in \mathbf{A}$, $I_{n+1}(a)$ is an answer set of $a^{In_{I_n}(a)}$. Since $I$ is the fixpoint, this becomes $I(a)$ is a model of $a^{In_I(a)}$. With Definition 8, this allows us to conclude that $I$ is a model of $F$.

THEOREM 4 *Let $F = \langle \mathbf{A}, C \rangle$ be a LPAS without cycles. An interpretation $M$ is an answer set of $F$ iff it is a model of $F$. of $F$.*

*Proof.* The "only-if" part follows immediately from Corollary 1. For the "if" part let $M$ be a model of $F$. Let $I_O$ be the interpretation yielding an output of $\emptyset$. Because $F$ is cycle-free there must agents without any incoming channels. Let $A_1$ be the set of these agents. Now we can construct an interpretation $I_1$ such that $I_1(a) = M(a)$ for each $a \in A_1$. For the remaining agents $b$, we make sure that $I_1(b)$ is a model of $b^{In_{I_0}(b)}$. Since $M$ is a model and because the agents of $A_1$ do not receive information from other, we know that $I_1(a)$ is an answer set of $a^{In_{I_0}(a)} = a^{In_M(a)}$. Now let $A_2$ be the set of agents that receive information only from agents in $A_1$. This is possible since $F$ does not contain cycles. With this we can construct an interpretation $I_2$ such that $I_2(a) = M(a)$ for each $a \in A_2$. For the remaining agents $b$, we make sure that $I_2(b)$ is a model of $b^{In_{I_1}(b)}$. Due to the construction of $I_2$, we have that for all $a \in A_2$ $a^{In_{I_1}(a)} = a^{In_M(a)}$. Due to $M$ being a model of $F$, we must have that $I_2(a)$ is an answer set of $a^{In_{I_1}(a)}$. We can continue this process until we constructed an interpretation $I_n$ that covers all agents. Clearly, $I_n = M$. Moreover, the sequence $I_0 \ldots I_n$ induces a evolution of which $I_n$ is the fixpoint. With Definition 9, this implies that $M$ is an answer set of $F$.

## A.3. LPAS AND GAME THEORY

### A.3.1. *Phase 1*

THEOREM 6 *Let $G = \langle N, H, P, (u_i)_{i \in N} \rangle$ be a finite extensive game with perfect information and let $P_N$ and $P_S$ be its corresponding OCLPs according to Definition 15. Then, $s^*$ is a Nash equilibrium (resp. subgame perfect equilibrium) for $G$ iff $s^*$ is a answer set of $P_N$ (resp. $P_S$).*

*Proof.* We will break down this proof into a seperate proof for Nash equilibria and one for subgame perfect equilibria

A.3.1.1. *Nash equilibria* Before we start the actual proof, we first demonstrate that every model for $P_N$ is also an answer set of it and that a model of $P_N$ is a strategy profile for $G$.

LEMMA 1. *Let $G = \langle N, H, P, (u_i)_{i \in N} \rangle$ be a finite extensive game with perfect information and let $G_n$ be its corresponding OCLP according to Definition 15. Then, $M$ is an answer set for $G_n$ iff $M$ is a model of $G_n$.*

*Proof.* The "only-if"-part follows immediately from Theorem 2.

For the "if"-part, let $M$ be a model of $G_n$. We need to demonstrate that $M$ is a minimal model of $G^M_{n,c}$. We first demonstrate that $M$ is a model of $G^M_{n,c}$. We need to show that every rule and every constraint in $G^M_{n,c}$ is satisfied by $M$. The choice logic program $G^M_{n,c}$ is obtained from $G^*_n$ (i.e. Definition 4) by deleting all defeated rules w.r.t. $M$. Since $M$ is a model of $G^M_{n,c}$, we know that for the remaining rules must hold that either $B \not\subseteq M$ or $|H_r \cap M| = 1$. This implies that the rule is satisfied. So, $M$ satisfies every rule in $G^M_{n,c}$ and, thus, $M$ is a model of $G^M_{n,c}$. So it remains to be shown that $M$ is also minimal. The construction of $G_n$ assures, by placing the choice rules in the most specific component and by forcing that every atom appears exactly once in such a choice rule, that these rules can never be defeated w.r.t. $M$. From Definition 4, we know that all these choice rules are also in $G^M_{n,c}$. Thus, for every $a \in M$, we have a rule $a \oplus A \leftarrow \ \in G^M_{n,c}$. Since $M$ is a model of $G^M_{n,c}$, we know that $A \cap M = \emptyset$. Removing $a$ from $M$, would result in an unsatisfied rule. This makes that every total interpretation $N$ with $N^+ \subset M^+$ cannot be a model for $G^M_{n,c}$. So, $M$ is a minimal model of $G^M_{n,c}$, which makes $M$ an answer set of $G_n$.

LEMMA 2. *Let $G = \langle N, H, P, (u_i)_{i \in N} \rangle$ be a finite extensive game with perfect information, let $G_n$ be the corresponding OCLP and let $M$ be a model of $G_n$. Then, $M$ is a strategy profile for $G$.*

*Proof.* This follows immediately from the construction of $G_n$. The heads of the choice rules in the component $C^t$ correspond exactly with all the sets $A(h)$, with $h$ a non-terminal history. Because $C^t$ is the most specific component and since every atom appears exactly once in the head of such a choice rule, we know that a choice rule $r$ cannot be defeated in $C^t$ w.r.t. $M$. Since $M$ is a model, we know that $|H_r \cap M| = 1$. So, for every non-terminal history $h$, we have $A(h) \cap M = 1$. This makes that $M$ is a strategy profile for $G$.

Now that we have proven the above lemmas, we can proceed with demonstrating the equivalence between answer sets and Nash equilibria.

For the "if"-part, let $M$ be a Nash equilibrium for $G$. With Lemma 1, we need to prove that $M$ is a model of $G_n$ in order to conclude that $M$ is an answer set of $G_n$. We have two types of rules in our program: choice rules (e.g. rules with more than one head atom) and rules with a single head atom. Let us proceed with the former. The head of such a choice rule corresponds to $A(h)$ for some

non-terminal history $h$. Since Nash equilibria are also strategy profiles, we have that $|A(h) \cap M| = 1$. This implies that all our choice rules are applied and thus satisfied w.r.t. $M$.
Now consider the rules with a single head atom. Let $r$ be such rule with $B_r \subseteq M$ and $H_r = a \notin M$. Since $M$ is a Nash equilibrium, we must have that the player, player $i$, who has to make a decision took an action $b$ which would lead him to an outcome that gives a payoff at least as good as the outcome which involves the action $a$. The construction of $P_N$ guarantees that $a$ is a head atom of a choice rule $r'$ with $b \in H_{r'}$. This rule is trivially applicable, thus $b \in \Omega^M_{c(r)}(a)$. Because $M$ is a strategy profile, the other players will have actions in their strategy to deal with the possibility that player $i$ might have chosen $b$ instead of $a$. These actions are all contained in $M$. Filling in actions for player $i$, including $b$, this yields a terminal history $h'$. According to the creation of $P_N$, this results in a rule $r'' : b \leftarrow [h'']$ with $h''$ those actions from $h'$ which are not chosen by player $i$. This rule $r''$ is applied and since $h'$ results in a payoff at least as good as the one of $h$, we can use $r''$ to defeat $r$. Thus, $r$ is satisfied w.r.t. $M$. So we can conclude that $M$ is a model of $P_N$ and thus, with Lemma 1, an answer set of $P_N$ .
For the "only-if"-part, let $M$ be an answer set of $P_N$. From Lemma 2, we already know that $M$ is a strategy profile for $G$. So it remains to be shown that $M$ is a Nash equilibrium. This means that a player $i$ cannot, given the other players' actions, leave the equilibrium to obtain a better payoff. Or that all terminal histories containing for the other players actions from $M$ and possibly different actions for $i$ should not produce a better outcome for player $i$. For all actions $a \in M$ chosen by player $i$, we have that all the alternatives $b$ of $a$ cannot be in $M$. This implies, for all rules $r$ with $b \in H_r$, that $r$ is not applicable or defeated. When $r$ is not applicable, we know that none of the terminal histories containing $b$ and $B_r$ can be in the way for concluding that $M$ is a Nash equilibrium. When $r$ is defeated, we know that there exists an applied rule $r'$ with $a \in H_{r'}$ such that $c(r) \not\prec c(r')$. This means that there is a terminal history containing $a$ which guarantees a payoff at least as good as the payoff you might get for using $b$. This means that there is no rational reason to replace $a$ with $b$. Since this is true for every action of $a$, we may conclude that $M$ is indeed a Nash equilibrium.

A.3.1.2. *Subgame perfect equilibria* Before we start the actual proof, we first demonstrate that every model for $P_S$ is also an answer set of it and that a model of $P_S$ is a strategy profile for $G$.

LEMMA 3. *Let $G = \langle N, H, P, (u_i)_{i \in N} \rangle$ be a finite extensive game with perfect information and $G_s$ be its corresponding OCLP according to Definition 15. Then, $M$ is an answer set for $G_s$ iff $M$ is a model of $G_s$.*

*Proof.* The "only-if"-part follows immediately from Theorem 2.
For the "if"-part, let $M$ be a model of $G_s$. We need to demonstrate that $M$ is a minimal model of $G^M_{s,c}$. We first demonstrate that $M$ is a model of $G^M_{s,c}$. We need to show that every rule and every constraint in $G^M_{s,c}$ is satisfied by $M$. The choice logic program $G^M_{s,c}$ is obtained from $G^*_s$ (i.e. Definition 4) by deleting all defeated rules w.r.t. $M$. Since $M$ is a model of $G_s$, we know that for the remaining rules either $B \nsubseteq M$ or $|H_r \cap M| = 1$ holds. This implies that the rule is satisfied in $G^M$. So, $M$ satisfies every rule, thus, $M$ is a model of $G^M_{s,c}$. So it remains to be shown that $M$ is also minimal. The construction of $G_s$ assures, by placing the choice rules in the most specific component and by forcing that every atom appears exactly once in such a choice rule, that these rules can never be defeated w.r.t. $M$. With Definition 4, we have that all these choice rules are also in $G^M_{s,c}$. Thus, for every $a \in M$, we have a rule $a \oplus A \leftarrow \ \in G^M_{s,c}$. Since $M$ is a model of $G^M_{s,c}$, we know that $A \cap M = \emptyset$. Removing $a$ from $M$, would result in an unsatisfied rule. This makes that every total interpretation $N$ with $N^+ \subseteq M^+$ cannot be a model for $G^M_{s,c}$. So, $M$ is a minimal model of $G^M_{s,c}$ which makes $M$ an answer set of $G_s$.

LEMMA 4. *Let $G = \langle N, H, P, (u_i)_{i \in N} \rangle$ be a finite extensive game with perfect information , $G_s$ be the corresponding OCLP and let $M$ be a model of $G_n$. Then, $M$ is a strategy profile for $G$.*
*Proof.* This immediately follows from the construction of $G_n$. The heads of the choice rules in the component $C^t$ correspond exactly with all the sets $A(h)$, with $h$ a non-terminal history. Because $C^t$ is the most specific component and since every atom appears exactly once in the head of such choice rule, we know that a choice rule $r$ cannot be defeated in $C^t$ wrt $M$. Since $M$ is a model, we know that $|H_r \cap M| = 1$. So, for every non-terminal history $h$, we have $A(h) \cap M = 1$. This makes that $M$ is a strategy profile for $G$.

Now that we proved the above lemmas, we can proceed with demonstrating the equivalence between answer sets and subgame perfect equilibria.
To prove the "if"-part, let $M$ be a subgame perfect equilibrium of $G$. With Lemma 3, it suffices to demonstrate that $M$ is a model of $P_S$. Because $M$ is a strategy profile, we know that $|A(h) \cap M| = 1$ for every non-terminal history $h \in H$. Exactly those $A(h)$ are responsible for the creation of the choice rules in $C^t$. This immediately implies that these rules are applied and, thus, satisfied. Now it remains to be shown that rules with a single head atom are also satisfied. Let $r$ be a rule such that $B_r \subseteq M$ and $H_r \notin B_r$. Let $a \in H_r$. The construction of $P_S$ guarantees that $a \in A(h)$ for some $h \in H$. $M$ being a subgame perfect equilibrium states that $a$ is not part of the outcome of the subgame $G|_h$. This implies the existence of a terminal history $h'$ of this subgame yielding a better or equal payoff for the player $P(h) = i$. The construction of $P_S$ makes

sure that a rule $r'' : b \leftarrow B$ exists with $B \subseteq M$, containing all the actions of $h'$ except the ones made by $i$. Clearly, $b \in M$ and $b \in \Omega^{M}_{c(r)}(a)$. Since the payoff for $i$ is at least as good as when $h$ is played, we have that $c(r') \preccurlyeq c(r)$. This means that $r'$ can be used as a defeater of $r$, which makes $r$ satisfied. So we may conclude that $M$ is indeed a model of $P_S$. With Lemma 3, we obtain that $M$ is also an answer set of $P_S$.
For the "only-if"-part, let $M$ be an answer set of $P_S$. Lemma 4 already tells us that $M$ is a strategy profile for $G$. When we look at the construction of $P_S$, we see subgames are imitated. The smallest subgames are represented by facts, since the actions that ought to be chosen are directly connected to the leafs. Rules belonging to the same subgame can possibly defeat each other. Since answer sets are models, for facts belonging to the same subgame, the fact which is preferred the most is made true. In case there are several, a random choice is justified. Clearly the chosen action corresponds to a Nash equilibrium of the subgame. The same applies for larger subgames. They are represented in the program in a similar way, only the number of body elements grows. Rules that become applicable contain only actions belonging to a Nash equilibrium of the lower subgames. For each larger subgame an action is chosen which yields the best or similar payoff to the deciding player. Clearly, in this way we have obtained a Nash equilibrium for the current subgame. When we reach the root of the original game $G$ in this way, we know that the collection of actions $M$ indeed contains a Nash equilibrium of each subgame in $G$. This makes that $M$ indeed is a subgame perfect equilibrium for $G$.

A.3.2. *Phase 2*

THEOREM 7 *Let $\langle N, H, P, (u_i)_{i\in N}\rangle$ be a finite extensive game with perfect information and let $P^p_N$ and $P^p_S$ be its corresponding OCLPs. Then, $s^*$ is a Nash equilibrium (resp. subgame perfect equilibrium) for $\langle N, H, P, (u_i)_{i\in N}\rangle$ iff $s^*$ is a answer set of $P^p_N$ (resp. $P^p_S$).*

*Proof.* The programs $P^p_N$ and $P^p_N$ are constructed in such a way that the relative order between the components of two rules with in their heads alternatives is the same as in $P_p$ and $P_S$. The only difference that we need to look into is the presence of choice rules in the most specific component of each player. With the construction of the programs $P^p_N$ and $P^p_S$ and the defeating strategy, we know that such a choice rule can only be defeated when more than one alternative is forced and decided on in the same component as the choice rule. When we compute the reduct in this case, we obtain that every rule containing an alternative generated by the choice rule has been deleted, including the choice rule itself. This means that a stable model of this reduct can never contain any of these alternatives, which makes that the interpretation at hand can never be considered an answer set. This implies that every answer set contains exactly one action for each decision some agents has to make. Thus, answer sets of $P_N^p$ and $P_S^p$ are strategy profiles. With this in mind, we have that our proof is the same as the proofs for demonstrating that answer sets of $P_N$ and $P_S$ coincide with respectively the Nash equilibria and subgame perfect equilibria of the represented game. (Theorems 6 and 6 on page 21).

A.3.3. *Phase 3*

THEOREM 8 *Let* $\langle N,H,P,(u_i)_{i\in N}\rangle$ *be a finite extensive game with perfect information and let* $S^N$ *and* $S^S$ *be the corresponding LPAS, according to Definition 17. Then,* $s^*$ *is a Nash equilibrium (resp. subgame perfect equilibrium) for* $\langle N,H,P,(u_i)_{i\in N}\rangle$ *iff the interpretation* $I$ *with* $I(a) = s^*$ *for every* $a \in \mathbf{A}$ *is a answer set of* $S^N$ *(resp.* $S^S$*).*

*Proof.* The construction of framework makes sure that a decision made by one agent is accepted by all the others. The reason for this is that atoms are considered to be alternatives by one agent only appear in the head of a rule of just this agent. With this in mind it is easy to see every agent has the same output in model and that the models of $(S^N)$ (resp. $S^S$) coincide with the answer sets of $P_N^p$ (resp. $P_s^n$). In theorem 7, we demonstrated that the set of answer sets of $P_N^p$ (resp. $P_S^p$) equals the set of Nash equilibria (resp. subgame perfect equilibria) of the represented game. Now, it remains to shown that every model is also an answer set. The reverse follows immediately from Corollary 1. Let $M$ be the output produced by every agent. We will demonstrate that there exist a evolution $I_\emptyset,\ldots,I_n$ with $n$ the number of players in the game. In the second step we have:

$$I_1(A^i) = M \cap A_i \ ;$$

with $A_i$ the actions from with player/agent $A^i$ can choose. This is made possible because of the choice rules which are immediately applicable. All the other steps are necessary to notify the agents of the choices made by the other agents. This makes that the output of each agent in the last step (i.e. step $n$) equals $M$. Only in the last step, all non-choice rules can become applicable. This makes it easy to verify that $I_i$, for $i < n$ fulfills all requirements necessary to take part in a evolution. Also $I_n$ satisfies the criteria to take part in the evolution, this because $M$ is a model of $S^N$ (resp. $S^S$). Clearly for the same reason we have that $I_n$ is a fixpoint of this evolution. So, we may conclude that every answer set of $S^N$ (resp. $S^S$) is a Nash equilibrium (resp. subgame perfect equilibrium) and vice versa.

A.4. (EXTENDED) LOGIC PROGRAMS

THEOREM 9 *Let $P$ be an extended logic program without classical negation. Then, $M \subseteq \mathcal{B}_P$ is an answer set of $P$ iff $S$ is an answer set of $P_L$ with $S^+ = M^+ \cup \mathrm{not}_{(\mathcal{B}_P \setminus M)}$.*
To make the proof more readable, we introduce, for an interpretation $I$ of $P_L$, $I^p$ and $I^n$ as respectively $\{a \in \mathcal{B}_P \mid a \in I\}$ and $\{a \in \mathcal{B}_P \mid \mathrm{not}_a \in I\}$. Thus, $I^+ = I^p \cup \mathrm{not}_{I^n}$.

*Proof.* Let us start with the "only-if"-part. So let $M$ be an answer set of $P$. We need to show that $S$ is an answer set of $P_L$. With Definition 4, this implies that $S$ has to be a minimal model of $P_L^S$.

- $S$ is a model of $P_L^S$. Let us consider the rules in the various components.
  - $r : a \oplus \mathrm{not}_a \leftarrow \ a \in P_L^S$ . Assume that $a \in S$. This implies that $a \in M$. Because $M$ is an interpretation, $a \notin M^-$. With the construction of $S$, we obtain $\mathrm{not}_a \notin S$. Thus, $|H_r \cap S| = 1$, which makes the rule satisfied. If $a \notin S$, we immediately have that $r$ is satisfied.
  - $r : l \leftarrow B, \mathrm{not}_C \in P_L^S$ with $l = a$ or $l = \mathrm{not}_a$. This implies that $r' : l' \leftarrow B, \neg C \in P$ with $l' = a$ or $l' = \neg a$ respectively. Because $M$ is deductively closed (i.e. every applicable rule is applied) for $P$, we have for $r'$ that one of the following:
    * $B \not\subseteq M$. Because $B \subseteq \mathcal{B}_P$ and $M^+ = S^+ \cap \mathcal{B}_P$, we obtain $B \not\subseteq S$.
    * $C \cap M \neq \emptyset$. Since $C \subseteq \mathcal{B}_P$ and $M^+ = S^p$, this yields $C \cap S^p \neq \emptyset$. Because $S^p \cap S^N = \emptyset$, we have $B_r \not\subseteq S$.
    * $a \in M$ if $l' = a$ with $a \in \mathcal{B}_P$. This implies $a \in S$.
    * $a \notin M$ if $l' = \neg a$ with $a \in \mathcal{B}_P$. With the construction of $S$, this yields $\mathrm{not}_a \in S$.

    Thus, the rule $r$ is satisfied by $S$.
  - $r : \ \leftarrow B, \mathrm{not}_C \in R$. This means that the constraint $\leftarrow B, \neg C \in P$. Because $M$ is an answer set of $P$, we have that $M$ is deductively closed. This implies that either $B \not\subseteq M$ or $C \cap M \neq \emptyset$. In terms of $S$, this yields $B \not\subseteq S$ or $\mathrm{not}_C \not\subseteq S$. This immediately implies that $r$ is not applicable, and thus satisfied.
  - $r : \mathrm{not}_a \leftarrow \ \in P_L^S$ . Suppose that $\mathrm{not}_a \notin S$. The construction of $S$ from $M$, implies $a \notin M^-$. Because $M$ is total, we have $a \in M^+$ and also $a \in S$. With the rule $a \oplus \mathrm{not}_a \leftarrow a \in C$, we obtain that $\Omega_N^S(\mathrm{not}_a) = \{a\}$. Because $M$ is an answer set $P$, we must have that $M$ is minimally closed for $P^M$. This implies:

$$\exists r' \in P \cdot a \in H_{r'}, B_{r'}^+ \subseteq M^+, B_{r'}^- \subseteq M^- \quad .$$

The construction of $P_L$ and $S$ supply us, in this case, with a rule $r''$ for which $a = H_{r''}$, $B_{r''} \subseteq S$ holds. Thus, by Definition 3, $r''$ defeats the rule $r$, as $c(r'') = R \prec c(r) = N$ and $a \in S$, which is in contradiction with $r \in P_L^S$.
So we have to conclude that $\mathrm{not}_a$ has to be in $S$, which makes $r$ satisfied.

So we can conclude that every rule in $P_L^S$ is satisfied w.r.t. $S$.

- $S$ is minimal. We will demonstrate that no total interpretation $Z$ with $Z^+ \subseteq S^+$ can be considered to be a model of $P_L^S$. Suppose that there exists a $\mathrm{not}_a$ such that $\mathrm{not}_a \in S$ while $\mathrm{not}_a \notin Z$. If $\mathrm{not}_a \in S$, we must have that $a \notin S$. Otherwise, the rule $a \oplus \mathrm{not}_a \leftarrow a$ would not be satisfied, which is impossible as $S$ is a model of $P_L^S$. This makes that $\Omega_N^S(\mathrm{not}_a) = \emptyset$. This implies that the rule $\mathrm{not}_a \leftarrow \; \in N$ cannot be defeated w.r.t. $S$. Thus, $\mathrm{not}_a \leftarrow \; \in P_L^S$. So, $Z$ cannot be a model of $P_L^S$, as $\mathrm{not}_a \notin Z$.
So we must have that $S^N = Z^n$. Thus, $S^+ \setminus Z^+ = D$ with $D \subseteq \mathcal{B}_P$. Since $D \subseteq S^+$, we have that $D \subseteq M$. Because $M$ is an answer set of $P$, we know that $M$ is minimally closed under $P^M$. This implies that, since $D \subseteq M$, there is an atom $a \in D$ such that a rule $r \in P^M$ exists for which, amongst other things, that $B_r \subseteq M$ and $B_r \cap D = \emptyset$ holds. This implies $B_r \subseteq (M^+ \setminus D)$. This rule could only be in $P^M$ when for its corresponding rule $r'$ holds $B_{r'}^- \cap M = \emptyset$. Since $S^N = Z^n$, we obtain with respect to $Z$, $B = B_{r'}^+ \subseteq Z$ and $\mathrm{not}_C = \mathrm{not}_{B_{r'}^-} \subseteq Z$. Due to the construction of $P_L$, there is a rule $r'' : a \leftarrow B, \mathrm{not}_C \in P_L$. Since $a \in S$, we know by the construction of $S$, that $\mathrm{not}_a \notin S$. Because $\mathrm{not}_a$ is the only alternative of $a$, we have, with Definition 3, that $r''$ cannot be defeated w.r.t. $S$, which implies that $r'' \in P_L^S$. This makes that $Z$ is not a model, as $a \notin Z$ while $B_{r''} \subseteq Z$.
So we may conclude that, $S$ is a minimal model of $P_L^S$.

$S$ being a minimal model of the choice logic program $P_L^S$ yields, that $S$ is a stable model of $P_L^S$. This implies that $S$ is indeed an answer set of $P_L$.
For the "if"-part, let $S$ be an answer of $P_L$. Let $M = S^p \cup \neg S^N$. According to the definition of an answer set ([19]), we need to demonstrate that $M$ is minimally closed for $P^M$. Before doing this, we demonstrate that $S^p \cup S^N = \mathcal{B}_P$. Because $S$ is an answer set of $P_L$, it is also a model of $P_L$ (e.g. Theorem 2). This implies that all the rules in the component $C$ need to be satisfied. In case $a \in S^p$, we have that the rule $a \oplus \mathrm{not}_a \leftarrow a$ is applicable and not defeated w.r.t. $S$. Since $S$ is a model we must have that $\mathrm{not}_a \notin S$. In case $a \notin S^p$, we have that $\Omega_N^S(\mathrm{not}_a) = \emptyset$. This implies that the rule $\mathrm{not}_a \leftarrow$ cannot be defeated w.r.t. $S$. Thus, since $S$ is a model of $P_L$, $\mathrm{not}_a \in S$. This yields $S^p \cap S^N = \emptyset$ and $S^p \cup S^N = \mathcal{B}_P$.

- $M$ is closed for $P^M$. We need to prove that for every rule $r \in P^M$ holds that $H_r \in M$ whenever $B_r \subseteq M$. There are two types of rules: constraints and rules with a single head atom. Let us first start with the constraints. According to the definition of the reduct, we have that a rule $r'$ exists from which $r$ is originated. For $r'$ holds that $A = H^-_{r'} \subseteq M$ and $C = B^-_{r'} \cap M = \emptyset$. This yields $A \cap S^N = \emptyset$ and $C \subseteq S^N$. The construction of $P_l$ implies the existence of a rule $\mathrm{not}_A \leftarrow B_r, \mathrm{not}_C$. Because $S$ is an answer set of $P_L$ and, by Theorem 2 a model of $P_L$, we must have that $B_r \not\subseteq S$. Since $B_r \subseteq \mathcal{B}_P$, this implies that $B_r \not\subseteq M$, which makes $r$ satisfied. Let us now proceed with the rules without empty-head and let $r$ be such a rule. Assume that $B_r \subseteq M$. Since $r \in P^M$, we have a rule $r' : a \leftarrow B_r, \neg C \in P$ such that $B_r \subseteq M$ and $C \cap M = \emptyset$. With the creation of $M$, this implies $B_r \subseteq S^p$ while $C \subseteq S^N$. Thus, $B \cup \mathrm{not}_C \subseteq S$. The construction of $P_L$ turns the rule $r'$ into a rule $r'' : a \leftarrow B_r, \mathrm{not}_C \in R$ with $B_{r''} \subseteq S$. Such a rule can never be defeated and since $S$ is a model of $P_L$, we must have $a \in S^p$. This yields, $a \in M$. Thus, $r$ is satisfied w.r.t. $M$. We showed that every rule in $P^M$ is satisfied, so we may conclude that $M$ is indeed closed.

– $M$ is minimally closed for $P^M$. We will prove that a set of atoms $Z$ for $P^M$ with $Z \subset M$ can never be closed under $P^M$. Because $S$ is an answer set of $P_L$, we have that $S$ is also a stable model of $P_L^M$. This means, by Proposition 6 in [10], that $S$ is unfounded-free. This implies that no non-empty subset of $S^+$ can be an unfounded set of $P_L$ w.r.t. $S$. Since $S^p = M^+$, we have that $D = (M^+ \setminus Z^+) \subseteq S^+$ with $D \neq \emptyset$. This means that $D$ cannot be an unfounded set w.r.t. $S$. According to the definition ([10]), this implies, amongst other things, that:

$$\exists a \in D \cdot \exists r : A \leftarrow B, \mathrm{not}_C \in P_L^S \cdot a \in H_r, B_r \subseteq S, B_r \cap D = \emptyset \ .$$

The last condition implies that $r$ must be a rule in the component $R$. The construction of $P_L$ implies that a rule $r' : a \leftarrow B, \neg C \in P$ exists such that $B \subseteq (M^+ \setminus D) = Z^+$ and $C \cap M = \emptyset$. This yields $r'' : a \leftarrow B \in P^M$ such that $a \notin Z$ and $B \subseteq Z$. Thus, $Z$ cannot be a model of $P^M$, which makes $M$ minimal.

We have demonstrated that $M$ is indeed a minimal closed set under $P^M$. Therefore, we can conclude that $M$ is an answer set of $P$.